Twido*Suite* V1.2

Guide de programmation

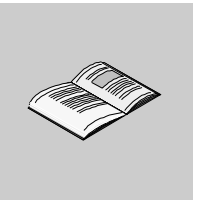
03/2007

a brand of Schneider Electric

Telemecanique

www.telemecanique.com

Table des matières



Consignes de sécurité 11

A propos de ce manuel 15

Partie I Description du logiciel Twido 17

Chapitre 1 Introduction au logiciel Twido 19

Introduction à TwidoSuite 20
Introduction aux langages Twido 21

Chapitre 2 Objets langage Twido 25

Validation d'un objet langage 26
Objets bits 27
Objets mots 29
Objets flottants et mots doubles 33
Adressage d'objets bits 37
Adressage d'objets mots 38
Adressage d'objets flottants 39
Adressage d'objets mots doubles 40
Repérage des entrées/sorties 41
Network Addressing 43
Objets blocs fonction 44
Objets structurés 46
Objets indexés 49
Symbolisation d'objets 51

Chapitre 3 Mémoire utilisateur 53

Structure de la mémoire utilisateur 54
Backup et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue 58
Backup et restauration avec une cartouche de sauvegarde de 32 Ko 60
Utilisation de la cartouche de mémoire étendue 64 Ko 63

Chapitre 4 Gestion des tâches événementielles 65

Vue d'ensemble des tâches événementielles 66
Description des différentes sources d'événement 67
Gestion des événements 69

Partie II Fonctions spéciales 71

Chapitre 5 Communications 73
Présentation des différents types de communication 74
Communication entre TwidoSuite et l'automate 76
Communication entre TwidoSuite et un modem 83
Communications par liaison distante 96
Communications ASCII 110
Communications Modbus 123
Requêtes Modbus standard 142
Classe d'implémentation Transparent Ready (Twido série A05, Ethernet A15) 148

Chapitre 6 Fonctions analogiques intégrées 149
Point de réglage analogique 150
Voie analogique 152

Chapitre 7 Gestion des modules analogiques 153
Vue d'ensemble des modules analogiques 154
Adressage d'entrées et de sorties analogiques 155
Configuration d'entrées et de sorties analogiques 156
Informations sur l'état du module analogique 163
Exemples d'utilisation de modules analogiques 165

Chapitre 8 Configuration d'entrée/sortie Twido Extreme 167
8.1 Introduction aux entrées et aux sorties Twido Extreme 168
Adressage des entrées/sorties (E/S) Twido Extreme 168
8.2 Configuration des entrées Twido Extreme 171
Configuration des entrées TOR de Twido Extreme 172
Configuration d'entrée analogique Twido Extreme 178
Configuration d'entrée PWM Twido Extreme 182
Exemple de configuration d'entrée PWM de Twido Extreme 184
8.3 Configuration des sorties de Twido Extreme 190
Configuration des sorties TOR de Twido Extreme 191
Configuration de la sortie du générateur d'impulsions (PLS) Twido Extreme 194
Configuration de la sortie PWM en mode Standard de Twido Extreme 200
Configuration de la sortie PWM en mode Hydraulique de Twido Extreme 209
Exemple de configuration de la sortie PWM hydraulique de Twido Extreme 216

Chapitre 9 Mise en œuvre du bus AS-Interface V2 219
Présentation du bus AS-Interface V2 220
Description fonctionnelle générale 221
Principes de mise en œuvre logicielle 224
Description de l'écran de configuration du bus AS-Interface 225
Configuration du bus AS-Interface 227
Description de la fenêtre AS-Interface en mode connecté 233
Modification de l'adresse d'un esclave 236
Mise à jour de la configuration du bus AS-Interface en mode connecté 238
Adressage automatique d'un esclave AS-Interface V2 242

Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante 243
Remplacement automatique d'un esclave AS-Interface V2 défectueux 244
Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2 245
Programmation et diagnostic du bus AS-Interface V2 246
Mode de fonctionnement du module interface bus AS-Interface V2 250

Chapitre 10 Installation et configuration du bus de terrain CANopen . . . 251

10.1 Présentation du bus de terrain CANopen 252
Base de connaissances CANopen 253
A propos de CANopen 254
Boot-Up CANOpen 257
Transmission de Process Data Object (PDO) 260
Accès aux données à l'aide d'échanges explicites (SDO) 262
"Node Guarding" et "Life Guarding" 263
Gestion du bus interne 265
10.2 Mise en œuvre du bus CANopen 266
Vue d'ensemble 267
Configuration matérielle 269
Méthode de configuration 270
Déclaration d'un module maître CANopen 272
Configurateur CANopen 273
Déclaration des équipements esclaves CANopen sur le réseau 275
Mappage des objets CANopen 279
Liaison des objets CANopen 283
Symbolisation des objets CANopen 285
Adressage des PDO du module maître CANopen 287
Programmation et diagnostic du bus de terrain CANopen 289
Echange sous tension CANopen pour les automates Twido 296

Chapitre 11 Installation et configuration du bus de terrain CANJ1939 . . . 297

11.1 Vue d'ensemble du bus de terrain CANJ1939 298
Base de connaissances CANJ1939 299
Numéro de groupe de paramètres et numéro de paramètre suspect CANJ19139 301
Identifiant CANJ1939 302
Communication sur un réseau CANJ1939 304
11.2 Mise en œuvre du bus CANJ1939 305
Vue d'ensemble de la mise en œuvre du CANJ1939 306
Configuration matérielle 307
Méthode de configuration du CANJ1939 308
Boîtes de dialogue de configuration CANJ1939 (Elément, Réseau, Port) 310
Création ou suppression d'objets Transmettre/Recevoir CANJ1939 314
Affichage des objets Transmettre/Recevoir CANJ1939 322
Configuration de la diffusion pour le CANJ1939 325
Configuration poste à poste du CANJ1939 327
Configuration du CANJ1939 en mode Expert 329
Objets d'entrée/sortie CANJ1939 330
Requête d'une sortie PGN 333

Chapitre 12 Configuration de la passerelle Ethernet TwidoPort 337
12.1 Configuration normale et connexion du module TwidoPort 338
Configuration normale avec TwidoSuite 339
Configuration BootP 347
12.2 Configuration Telnet de TwidoPort 348
Présentation de la configuration Telnet 349
Menu principal Telnet 350
Paramètres IP/Ethernet 351
Configuration des paramètres série 353
Configuration de la passerelle 354
Configuration de la sécurité 355
Statistiques Ethernet 356
Serial Statistics 357
Enregistrement de la configuration 358
Restauration des paramètres par défaut 359
Mise à niveau du microprogramme TwidoPort 360
Vous avez oublié votre mot de passe et/ou votre configuration IP ? 362
12.3 Fonctions de communication 363
Fonctionnalités Ethernet 364
Protocole de communication Modbus/TCP 365
Codes de fonction Modbus pris en charge localement 366

Chapitre 13 Fonctionnement de l'afficheur 367
Afficheur 368
Informations d'identification et états de l'automate 371
Variables et objets système 373
Paramètres de port série 379
Horloge calendaire 380
Facteur de correction de l'horodateur 381

Partie III Description des langages Twido 383

Chapitre 14 Langage schéma à contacts 385
Introduction aux schémas à contacts 386
Principes de programmation en langage schéma à contacts 388
Blocs de schémas à contacts 390
Eléments graphiques du langage schéma à contacts 393
Instructions spéciales OPEN et SHORT du langage schéma à contacts 396
Conseils de programmation 398
Réversibilité schéma à contacts/liste d'instructions 401
Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions 403
Documentation du programme 405

Chapitre 15 Langage liste d'instructions 407
Vue d'ensemble des programmes en langage liste d'instructions. 408
Fonctionnement des listes d'instructions. 410
Instructions en langage liste d'instructions 411
Utilisation de parenthèses. 414
Instructions de pile (MPS, MRD, MPP) 417

Chapitre 16 Grafcet 419
Description des instructions Grafcet 420
Description de la structure d'un programme Grafcet. 425
Actions associées aux étapes Grafcet 429

Partie IV Description des instructions et des fonctions 431

Chapitre 17 Instructions élémentaires 433
17.1 Traitement booléen 434
Instructions booléennes 435
Explication du format de description des instructions booléennes 437
Instructions de chargement (LD, LDN, LDR, LDF) 439
Instructions d'affectation (ST, STN, R, S) 441
Instructions AND logique (AND, ANDN, ANDR, ANDF) 443
Instructions OR logique (OR, ORN, ORR, ORF). 445
OR exclusif, instructions (XOR, XORN, XORR, XORF) 447
Instruction NOT (N) 449
17.2 Blocs fonctions élémentaires. 450
Blocs fonctions standard 451
Principes de programmation de blocs fonction standard 453
Bloc fonction temporisateur (%TMi) 455
Type de temporisateur TOF 457
Type de temporisateur TON 458
Type de temporisateur TP. 459
Programmation et configuration de temporisateurs 460
Bloc fonction compteur/décompteur (%Ci) 463
Programmation et configuration des compteurs 467
Bloc fonction registre bits à décalage (%SBRi). 470
Bloc fonction pas à pas (%SCi). 472
17.3 Traitement numérique 474
Introduction aux instructions numériques 475
Instructions d'affectation 476
Instructions de comparaison 481
Instructions arithmétiques sur entiers 483
Instructions logiques 487
Instructions de décalage 488
Instructions de conversion. 490
Instructions de conversion entre mots simples et doubles 492

17.4 Instructions du programme 493
Instructions END 494
Instruction NOP 496
Instructions de saut 497
Instructions de sous-programme 498

Chapitre 18 Instructions avancées 501

18.1 Blocs fonctions avancés 503
Objets mots et objets bits associés à des blocs fonction avancés 504
Principes de programmation de blocs fonctions avancés 506
Bloc fonction registre LIFO/FIFO (%Ri) 509
LIFO, fonctionnement 510
Fonctionnement FIFO 511
Programmation et configuration des registres 512
Bloc fonction %PWM (modulation de la largeur d'impulsion) 514
Bloc fonction sortie du générateur d'impulsions (%PLS) 517
Bloc fonction programmateur cyclique (%DR) 520
Fonctionnement du bloc fonction programmateur cyclique %DRi 521
Programmation et configuration des programmateurs cycliques 523
Bloc fonction compteur rapide (%FC) 525
Bloc fonction compteur rapide (%VFC) 528
Emission/réception de messages – Instruction d'échange (EXCH) 540
Bloc fonction de contrôle d'échange (%MSGx) 541
18.2 Fonctions horodateur 545
Fonctions horloges 546
Blocs horodateurs 547
Horodatage 549
Réglage de la date et de l'heure 551
18.3 Guide de démarrage rapide de l'automate PID Twido 554
Objectif du document 555
Etape 1 - Configuration des voies analogiques utilisées pour le contrôle 557
Etape 2 - Conditions préalables à la configuration de l'automate PID 559
Etape 3 - Configuration du PID 561
Etape 4 - Initialisation de la configuration du contrôle 567
Etape 5 - Fonction AT + PID pour l'installation du contrôle 571
Etape 6 - Mise au point des réglages 575
18.4 Fonction PID 577
Vue d'ensemble 579
Principe de la boucle de régulation 580
Méthodologie de développement d'une application de régulation 581
Compatibilités et performances 582
Caractéristiques détaillées de la fonction PID 583
Comment accéder à la configuration du PID 586
Eléments PID de la fonction PID 587
Onglet Général de la fonction PID 591
Onglet Entrée de la fonction PID 594

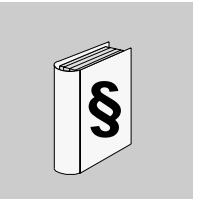
Onglet PID de la fonction PID 596
Onglet AT de la fonction PID. 598
Onglet Sortie de la fonction PID 603
Comment accéder à la mise au point du PID 606
Onglet Animation de la fonction PID 607
Ecran Trace de la fonction PID 609
Etats du PID et codes d'erreurs 611
Réglage PID avec la fonction d'auto tuning (AT) 615
Méthode de réglage du paramètre PID. 623
Rôle et influence des paramètres PID. 626
Annexe 1 : Notions fondamentales de la théorie PID 630
Annexe 2 : Premier ordre avec modèle de temporisation 632
18.5 Instructions sur flottants 634
Instructions arithmétiques sur flottant 635
Instructions trigonométriques 639
Instructions de conversion. 641
Instructions de conversion Entier <-> Flottant. 642
18.6 Instructions ASCII 645
Instruction ROUND 646
Conversion d'une valeur ASCII en valeur entière 648
Conversion d'une valeur entière en valeur ASCII 650
Conversion d'une valeur ASCII en flottant 652
Conversion d'un flottant en valeur ASCII 654
18.7 Instructions sur tableaux d'objets 656
Fonction de sommation sur tableaux 657
Fonction de comparaison de tableaux 659
Fonctions de recherche sur tableaux 661
Fonctions de recherche de valeurs maximum et minimum sur tableaux 663
Nombre d'occurrences d'une valeur dans un tableau 664
Fonction décalage circulaire sur un tableau 665
Fonction de tri sur tableau 667
Fonction d'interpolation sur tableau de flottants 668
Fonction de moyenne des valeurs d'un tableau de flottants 672

Chapitre 19 Bits système et mots système 673
Bits système (%S) 674
Mots système (%SW) 683

Glossaire 697

Index 711

Consignes de sécurité



Informations importantes

AVIS

Lisez attentivement ces instructions et familiarisez-vous avec le matériel avant d'essayer de l'installer, de le faire fonctionner ou d'effectuer une opération de maintenance. Les messages spéciaux qui suivent peuvent apparaître partout dans ce document ou sur l'appareil. Ils vous avertissent de dangers potentiels ou attirent votre attention sur des renseignements pouvant éclairer ou simplifier une procédure.

L'apposition de ce symbole à un panneau de sécurité Danger ou Avertissement signale un risque électrique pouvant entraîner des lésions corporelles en cas de non-respect des consignes.

Ceci est le symbole d'une alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

⚠ DANGER

DANGER indique une situation immédiatement dangereuse qui, si elle n'est pas évitée, **entraînera** la mort ou des blessures graves.

⚠ AVERTISSEMENT

AVERTISSEMENT indique une situation présentant des risques susceptibles de **provoquer** la mort, des blessures graves ou des dommages matériels.

⚠ ATTENTION

ATTENTION indique une situation potentiellement dangereuse et susceptible d'**entraîner** des lésions corporelles ou des dommages matériels.

REMARQUE

L'entretien du matériel électrique ne doit être effectué que par du personnel qualifié. Schneider Electric ne saurait être tenu responsable des conséquences éventuelles découlant de l'utilisation de cette documentation. Ce document n'est pas destiné à servir de manuel d'utilisation aux personnes sans formation. Les instructions d'assemblage et d'installation sont fournies dans le Guide de référence du matériel d'automates programmables Twido (pour bases compactes et modulaires, bases Twido Extreme, modules d'E/S TOR, modules d'E/S analogiques ou modules de communication).

(c) 2002-2007 Schneider Electric Tous droits réservés

Informations supplémentaires relatives à la sécurité

Les personnes chargées de l'application, de la mise en œuvre ou de l'utilisation de ce produit doivent s'assurer que les principes de conception fondamentaux ont été inclus dans chacune des applications, en totale conformité avec les normes, codes, règlements, exigences en matière de performance et de sécurité et lois en vigueur.

Avertissements généraux et précautions à prendre

⚠ DANGER

RISQUE D'ELECTROCUTION, D'INCENDIE OU D'EXPLOSION

Coupez l'alimentation avant de commencer l'installation, le retrait, le câblage, la maintenance ou le contrôle du système à relais intelligent.

Le non-respect de cette directive entraînera la mort, des blessures graves ou des dommages matériels.

⚠ AVERTISSEMENT

RISQUE D'EXPLOSION

- Le remplacement des composants risque d'affecter la conformité de l'équipement à la Classe 1, Division 2.
- Assurez-vous que l'alimentation est coupée ou que la zone ne présente aucun danger avant de déconnecter l'équipement.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

⚠ AVERTISSEMENT

FONCTIONNEMENT ACCIDENTEL DE L'EQUIPEMENT

- Coupez l'alimentation avant de procéder au retrait, à l'installation, au câblage ou à l'entretien.
- Ce produit n'est pas conçu pour être utilisé lors d'opérations dangereuses pour la sécurité. Lorsque des risques de lésions corporelles ou de dommages matériels existent, utilisez les verrous de sécurité appropriés.
- Les modules ne doivent être ni démontés, ni réparés, ni modifiés.
- Cet automate est conçu pour être utilisé dans un coffret.
- Installez les modules dans des conditions de fonctionnement normales.
- L'alimentation des capteurs doit uniquement servir à alimenter les capteurs connectés au module.
- Pour les circuits d'alimentation et de sortie, utilisez un fusible conçu conformément aux standards de type T de la norme CEI60127. Le fusible doit répondre aux exigences de courant et de tension du circuit. Fusibles recommandés : fusibles série 218 Littelfuse® 5 x 20 mm à action retardée.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Mise au rebut de la pile

Les bases compactes TWDLCA•40DRF utilisent une pile lithium externe en option permettant de prolonger la durée de stockage des données. (Remarque : La pile lithium n'est pas fournie avec les bases compactes, vous devez l'acheter séparément.)

⚠ AVERTISSEMENT

RISQUE D'EXPLOSION ET DE TOXICITE

- N'incinérez pas de pile lithium, car elle risque d'exploser et de générer des substances toxiques.
- Ne manipulez pas une pile lithium qui fuit ou qui est endommagée.
- Les piles épuisées doivent être mises au rebut de manière appropriée. Une mise au rebut inappropriée des piles non utilisées peut avoir des effets dangereux ou négatifs sur l'environnement.
- Dans certaines zones, la mise au rebut de piles lithium avec les ordures ménagères est interdite. Quoi qu'il en soit, vous êtes tenu de toujours vous conformer aux réglementations locales de votre région ou de votre pays en ce qui concerne la mise au rebut des piles.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Avertissement concernant l'inversion de polarité

La polarité inversée au niveau de la sortie transistor n'est pas autorisée.

Les sorties transistor des bases compactes TWDLCA•40DRF ne peuvent supporter aucune inversion de polarité.

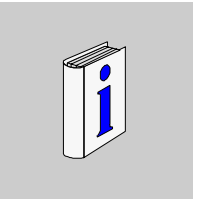
⚠ ATTENTION

RISQUE D'ENDOMMAGEMENT DES SORTIES TRANSISTOR EN RAISON DE L'INVERSION DE LA POLARITE

- Respectez les marques de polarité aux borniers des sorties transistor.
- Une inversion de polarité peut endommager définitivement ou détruire les circuits de sortie.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

A propos de ce manuel



Présentation

Objectif du document

Le manuel de référence du logiciel des automates programmables Twido est composé des sections suivantes :

- Description du logiciel de programmation Twido et introduction aux notions fondamentales requises pour programmer les automates Twido.
- Description des communications, de la gestion des E/S analogiques, de l'installation du module d'interface de bus AS-Interface et du module maître de bus de terrain CANopen, et d'autres fonctions spéciales.
- Description des langages logiciels utilisés pour créer des programmes Twido.
- Description des instructions et des fonctions des automates Twido.

Champ d'application

Les informations du présent manuel s'appliquent **uniquement** aux automates programmables Twido.

Les données et illustrations fournies dans cette documentation ne sont pas contractuelles. Nous nous réservons le droit de modifier nos produits conformément à notre politique de développement permanent. Les informations présentes dans ce document peuvent faire l'objet de modifications sans préavis et ne doivent pas être interprétées comme un engagement de la part de Schneider Electric.

Avertissements liés au(x) produit(s)

Schneider Electric ne saurait être tenu responsable des erreurs pouvant figurer dans le présent document. Merci de nous contacter pour toute suggestion d'amélioration ou de modification ou si vous avez trouvé des erreurs dans cette publication.

Aucune partie du présent document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique ou mécanique (photocopiage inclus), sans autorisation écrite expresse de Schneider Electric.

Copyright © Schneider Electric 2007. Tous droits réservés.

Toutes les réglementations de sécurité pertinentes locales doivent être observées lors de l'installation et de l'utilisation du présent produit. Pour des raisons de sécurité et pour garantir une conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque les automates sont utilisés pour des applications présentant des exigences de sécurité technique, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou du logiciel approuvé avec nos produits peut entraîner des blessures, des dommages ou des dysfonctionnements.

Le non-respect du présent avertissement relatif au produit peut entraîner des blessures ou des dommages matériels.

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail techpub@schneider-electric.com

Description du logiciel Twido

I

Présentation

Objet de cette partie

Cette rubrique fournit une introduction aux langages du logiciel, ainsi que les principales informations requises pour créer des programmes de régulation des automates programmables Twido.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
1	Introduction au logiciel Twido	19
2	Objets langage Twido	25
3	Mémoire utilisateur	53
4	Gestion des tâches événementielles	65

Introduction au logiciel Twido

1

Présentation

Objet de ce chapitre

Ce chapitre offre une introduction rapide à TwidoSuite, le logiciel de programmation et de configuration des automates Twido, ainsi qu'aux langages de programmation Grafcet, liste d'instructions ou schéma à contacts.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Introduction à TwidoSuite	20
Introduction aux langages Twido	21

Introduction à TwidoSuite

Introduction

TwidoSuite est un environnement de développement graphique complet permettant de créer, configurer et gérer des applications d'automatisation pour les automates programmables Twido de Telemecanique. TwidoSuite vous permet de créer des programmes avec différents types de langage (voir *p. 21*), puis de transférer l'application en vue de son exécution sur un automate.

TwidoSuite

TwidoSuite est un programme 32 bits Windows destiné aux ordinateurs (PC) fonctionnant sous le système d'exploitation Microsoft Windows 2000/XP Professionnel.

Principales fonctionnalités logicielles offertes par TwidoSuite :

- Interface utilisateur intuitive orientée projet
- Conception logicielle sans menu. Toutes les tâches et les fonctions d'une étape de projet sélectionnée sont disponibles à tout moment.
- Support de programmation et de configuration
- Communications avec l'automate
- Aide relative aux tâches simples et fournissant des liens utiles vers l'aide en ligne

Note : La liaison Automate-PC utilise le protocole TCP/IP. Il est obligatoire que ce protocole soit installé sur le PC.

Configuration minimale

La configuration minimale requise pour l'utilisation de TwidoSuite est la suivante :

- Ordinateur compatible PC avec
  - processeur Pentium 466 MHz ou supérieur recommandé
  - 128 Mo de RAM ou supérieure recommandée
  - 100 Mo d'espace disque disponible
- Système d'exploitation : Windows 2000 ou Windows XP :
  - Evitez le correctif 834707-SP1 (corrigé par le correctif 890175) et le correctif 896358, qui entraînent des problèmes d'affichage de l'aide en ligne.
  - Service Pack 2 ou supérieur recommandé. Téléchargeable sur le site www.microsoft.com.

Introduction aux langages Twido

Introduction

Un automate programmable lit des entrées, commande des sorties et résout une logique basée sur un programme. La création d'un programme d'un automate Twido consiste à écrire une série d'instructions rédigées dans un des langages de programmation Twido.

Languages Twido

Les langages suivants peuvent être utilisés pour créer des programmes d'automates Twido :

- Langage liste d'instructions :
  Un programme liste d'instructions est constitué d'une série d'expressions logiques, rédigées sous la forme d'une séquence d'instructions booléennes.
- Langage schéma à contacts :
  Un schéma à contacts est une représentation graphique d'une expression logique.
- Langage Grafcet :
  Le langage Grafcet est constitué d'une succession d'étapes et de transitions. Twido comprend les instructions liste Grafcet, mais pas les objets de représentation graphique Grafcet.

Les opérations de création et d'édition de programmes Twido à l'aide de ces langages de programmation peuvent être réalisées depuis un ordinateur personnel (PC).

Une fonctionnalité de réversibilité liste d'instructions / schéma à contacts vous permet de convertir un programme en langage liste d'instructions dans le langage schéma à contacts, et vice-versa.

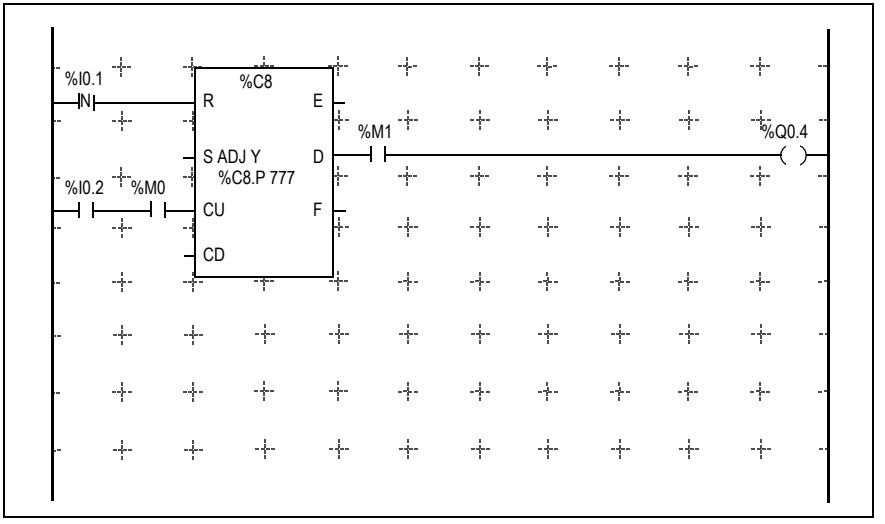
Langage liste d'instructions

Un programme rédigé en langage liste d'instructions consiste en une série d'instructions exécutées de manière séquentielle par l'automate. Vous trouverez ci-dessous un exemple de programme en langage liste d'instructions.

```
0   BLK     %C8
1   LDF     %I0.1
2   R
3   LD      %I0.2
4   AND     %M0
5   CU
6   OUT_BLK
7   LD      D
8   AND     %M1
9   ST      %Q0.4
10  END_BLK
```

Langage schéma à contacts

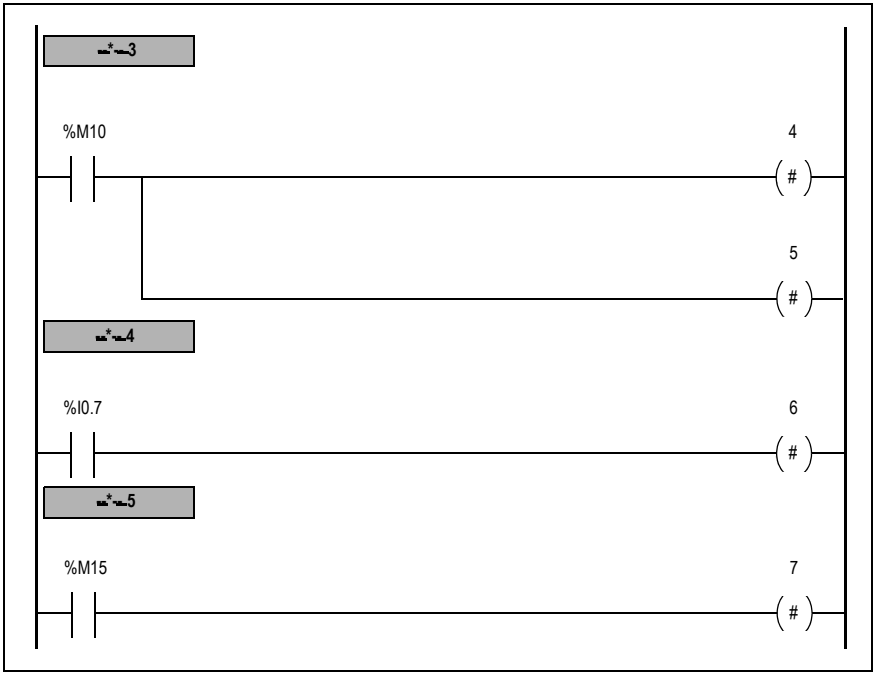
Les schémas à contacts utilisent la même représentation graphique que celle des circuits de relais en logique programmée. Dans ces schémas, les éléments graphiques, tels que des bobines, des contacts et des blocs représentent les instructions du programme. Ci-dessous un exemple de schéma à contacts.



Langage Grafcet

La méthode analytique Grafcet divise toute application d'automatisation en une série d'étapes auxquelles des actions, des transitions et des conditions sont associées. Vous trouverez ci-dessous des exemples d'instructions Grafcet, rencontrées respectivement dans des programmes liste d'instructions et schéma à contacts.

```
0     -*-     3
1     LD      %M10
2     #       4
3     #       5
4     -*-     4
5     LD      %I0.7
6     #       6
7     -*-     5
8     LD      %M15
9     #       7
10    ...
```



Objets langage Twido

2

Présentation

Objet de ce chapitre

Ce chapitre offre une description détaillée des objets langage de programmation des automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Validation d'un objet langage	26
Objets bits	27
Objets mots	29
Objets flottants et mots doubles	33
Adressage d'objets bits	37
Adressage d'objets mots	38
Adressage d'objets flottants	39
Adressage d'objets mots doubles	40
Repérage des entrées/sorties	41
Network Addressing	43
Objets blocs fonction	44
Objets structurés	46
Objets indexés	49
Symbolisation d'objets	51

Validation d'un objet langage

Introduction

Les objets mots et bits ne sont valides que lorsqu'un espace mémoire a été alloué dans l'automate. Pour que cette allocation soit possible, il est nécessaire que ces objets aient été utilisés dans l'application avant d'être téléchargés vers l'automate.

Exemple

La plage d'objets valides est comprise entre 0 et la référence maximum autorisée pour ce type d'objet. Par exemple, si la référence maximum autorisée pour les mots mémoire dans votre application est %MW9, les zones %MW0 à %MW9 sont allouées. Dans cet exemple, %MW10 n'est pas valide. Aucun accès à cette zone n'est autorisé, aussi bien de manière interne qu'externe.

Objets bits

Introduction

Les objets bits sont des variables logicielles de type bit qui peuvent être utilisés comme des opérandes et testés par des instructions booléennes. Vous trouverez ci-dessous la liste des objets bits :

- Bits d'E/S
- Bits internes (bits mémoire)
- Bits système
- Bits étape
- Bits extraits de mots

Liste des bits opérandes

Le tableau suivant répertorie et décrit les principaux objets bits qui sont utilisés comme opérandes dans des instructions booléennes.

Type	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Valeurs immédiates	0 ou 1 (False ou True)	0 ou 1	-	-
Entrées Sorties	Ces bits sont les "images logiques" des états électriques des E/S. Ils sont stockés dans la mémoire de données et sont mis à jour à chaque scrutation de la logique du programme.	%Ix.y.z (2) %Qx.y.z (2)	Remarque (3)	Non Oui
AS-Interface Entrées Sorties	Ces bits sont les "images logiques" des états électriques des E/S. Ils sont stockés dans la mémoire de données et sont mis à jour à chaque scrutation de la logique du programme.	 %IAx.y.z %QAx.y.z	Hinweis (4)	 Non Oui
Interne (mémoire)	Les bits internes sont des zones de mémoire internes utilisées pour stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution. **Remarque :** Les bits d'E/S non utilisés ne peuvent pas être employés comme des bits internes.	%Mi	128 TWDLC•A10DRF, TWDLC•A16DRF 256 Tous les autres automates	Oui
Système	Les bits système %S0 à %S127 surveillent le bon fonctionnement de l'automate ainsi que la bonne exécution du programme de l'application.	%Si	128	Selon i

Type	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Blocs fonction	Les bits des blocs fonction correspondent aux sorties des blocs fonction. Ces sorties peuvent être directement câblées ou exploitées en tant qu'objet.	%TMi.Q, %Ci.P, etc.	Remarque (3)	Non (5)
Blocs fonction réversibles	Blocs fonction programmés à l'aide d'instructions de programmation réversible BLK, OUT_BLK et END_BLK.	E, D, F, Q, TH0, TH1	Remarque (3)	Non
Extraits de mots	Pour certains mots, un des 16 bits est extrait en tant que bit opérande.	Variable	Variable	Variable
Etapes Grafcet	Les bits %X1 à %Xi sont associés aux étapes Grafcet. Le bit étape Xi est à l'état 1 lorsque l'étape correspondante est active et à l'état 0 lorsqu'elle est désactivée.	%Xi	62 TWDLC•A10DRF, TWDLC•A16 DRF 96 TWDLC•A24DRF, TWDLCA•40DRF et automates modulaires	Oui

Légendes :

1. Ecrit par le programme ou à l'aide de l'éditeur de table d'animation.
2. Reportez-vous à la section "Repérage des Entrées/Sorties".
3. Ce nombre est déterminé par le modèle de l'automate.
4. Où, x = adresse du module d'expansion (0..7); y = adresse AS-Interface (0A..31B); z = numéro de voie (0..3). (Voir *p. 284*.)
5. Ces bits, à l'exception de %SBRi.j et de %SCi.j, sont accessibles en écriture et en lecture.

Objets mots

Introduction

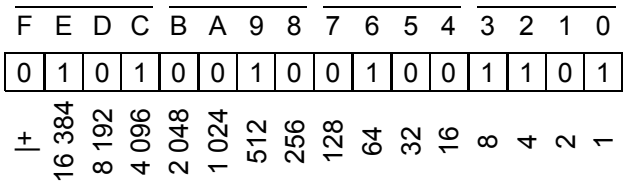
Les objets mots sont repérés sous la forme de mots de 16 bits rangés dans la mémoire de données et pouvant contenir un entier compris entre –32 768 et 32 767 (sauf pour le bloc fonction compteur rapide (FC) qui est compris entre 0 et 65 535).

Exemples d'objets mots :

- Valeurs immédiates
- Mots internes (%MWi) (mots mémoire)
- Mots constants (%KWi)
- Mots d'échange E/S (%IWi, %QWi%)
- Mots d'E/S analogiques AS-Interface (IWAi, %QWAi)
- Mots système (%SWi)
- Blocs fonction (données de configuration et/ou d'exécution)

Formats de mot

Le contenu des mots ou des valeurs est rangé dans la mémoire utilisateur sous la forme d'un code binaire à 16 bits (complément à deux) utilisant la convention suivante :



Pour les notations binaires signées, le bit 15 est attribué, par convention, au signe de la valeur codée :

- Le bit 15 est mis à 0 : le mot contient une valeur positive.
- Le bit 15 est mis à 1 : le mot contient une valeur négative (les valeurs négatives sont exprimées en complément à deux).

Il est possible d'entrer et de récupérer les mots et les valeurs immédiates sous les formats suivants :

- Décimal
  Min. : -32 768, Max. : 32 767 (1 579, par exemple)
- Hexadécimal
  Min. : 16#0000, Max. : 16#FFFF (16#A536, par exemple)
  Syntaxe alternative : #A536
- Les règles du format ASCII sont les suivantes :
  - La fonction lit toujours l'octet de poids fort en premier.
  - Tout caractère ASCII en dehors de l'intervalle ['0' - '9'] ([16#30 - 16#39]) est considéré comme un caractère de fin, à l'exception du signe moins '-' (16#2D) lorsqu'il est placé comme premier caractère.
  - En cas de débordement (>32 767 ou <-32 768), le bit système %S18 (erreur ou débordement arithmétique) est réglé sur 1 et la valeur 32 767 ou -32 768 est renvoyée.
  - Si le premier caractère de l'opérande est un caractère de "fin", la valeur 0 est renvoyée et le bit %S18 est réglé sur 1.

  Par exemple, "HELLO" :
  - `%MW0 := "HE"`
  - `%MW1 := "LL"`
  - `%MW2 := "O "`

Description des objets mots Le tableau suivant décrit les objets mots.

Mots	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Valeurs immédiates	Il s'agit d'entiers dont le format est identique à celui des mots de 16 bits. Cela permet d'attribuer des valeurs à ces mots.		-	Non
	Base 10	-32 768 à 32 767		
	Base 16	16#0000 à 16#FFFF		
Interne (mémoire)	Mots utilisés pour ranger des valeurs dans la mémoire des données au cours du fonctionnement.	%MWi	3 000	Oui
Constante	Mémorisent les constantes ou les messages alphanumériques. Leur contenu peut être écrit ou modifié uniquement à l'aide de TwidoSuite et en cours de configuration.	%KWi	256	Oui, uniquement à l'aide de TwidoSuite
Système	Ces mots de 16 bits comportent plusieurs fonctions : ● Ils permettent l'accès aux données provenant directement de l'automate en lisant les mots %SWi. ● Ils effectuent des opérations sur l'application (l'ajustement des blocs horodateurs, par exemple).	%SWi	128	Selon i
Blocs fonction	Ces mots correspondent aux paramètres ou aux valeurs courantes des blocs fonction.	%TM2.P, %Ci.P, etc.		Oui
Mots d'échange réseau	Attribués aux automates connectés en tant que Liaisons distantes. Ces mots sont utilisés pour la communication entre les automates :			
	Entrée réseau	%INWi.j	4 par liaison distante	Non
	Sortie réseau	%QNWi.j	4 par liaison distante	Oui
Mots d'E/S analogiques	Attribués aux entrées et sorties analogiques des modules esclaves AS-Interface.			
	Entrées analogiques	%IWAx.y.z	Remarque (2)	Non
	Sorties analogiques	%QWAx.y.z	Remarque (2)	Oui

Mots	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Bits extraits	Il est possible d'extraire un des 16 bits à partir des mots suivants :			
	Circuit interne	%MWi:Xk	1 500	Oui
	Système	%SWi:Xk	128	Dépend de i
	Constante	%KWi:Xk	64	Non
	Entrée	%IWi.j:Xk	Remarque (3)	Non
	Sortie	%QWi.j:Xk	Remarque (3)	Oui
	Entrée esclave AS-Interface	%IWAx.y.z:Xk	Remarque (3)	Non
	Sortie esclave AS-Interface	%QWAx.y.z:Xk	Remarque (3)	Oui
	Entrée réseau	%INWi.j:Xk	Remarque (3)	Non
	Sortie réseau	%QNWi.j:Xk	Remarque (3)	Oui

Note :
1. Ecrit par le programme ou à l'aide de l'éditeur de table d'animation.
2. Où x = adresse du module d'expansion (0..7), y = adresse AS-Interface (0A..31B), z = numéro de voie (0..3). (Voir *p. 284*.)
3. Ce nombre est déterminé par la configuration.

Objets flottants et mots doubles

Présentation

TwidoSuite permet d'effectuer des opérations sur les objets flottants et mots doubles entiers.

Un flottant est un argument mathématique qui possède une décimale dans son expression (exemples : 3,4E+38, 2,3 ou 1,0).

Un mot double entier est constitué de 4 octets stockés dans la mémoire de données et contenant une valeur comprise entre -2 147 483 648 et +2 147 483 647.

Format et valeur du flottant

Le format flottant utilisé est celui de la norme IEEE STD 734-1985 (équivalence CEI 559). La longueur des mots est de 32 bits, ce qui correspond à des nombres flottants simple précision.

Tableau représentant le format d'un flottant :

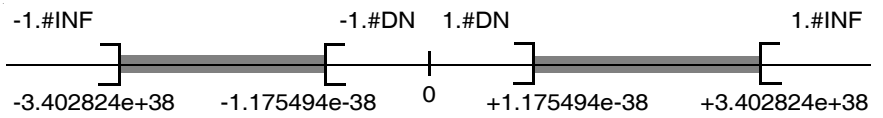
Bit 31	Bits {30...23}	Bits {22...0}
S	Exposant	Mantisse

La valeur du format ci-dessus est déterminée par l'équation suivante :

Valeur Flottant 32 bits = $(-1)^{S} * 2^{(\text{Exposant} - 127)} * 1,\text{Mantisse}$

Les valeurs flottantes peuvent être représentées avec ou sans exposant, elles doivent toujours comporter une virgule (virgule flottante).

Les valeurs flottantes sont comprises entre -3,402824e+38 à -1,175494e-38 et 1,175494e-38 à 3,402824e+38 (valeurs grisées sur le schéma). Elles comportent aussi la valeur 0 notée 0,0.



Lorsqu'un résultat de calcul est :

- inférieur à -3,402824e+38, le symbole -1.#INF (pour -infini) est affiché.
- supérieur à +3,402824e+38, le symbole 1.#INF (pour + infini) est affiché.
- compris entre -1,175494e-38 et 1,175494e-38, il est arrondi à 0,0. Une valeur comprise entre ces bornes ne peut être saisie en valeur flottante.
- indéfini (par exemple, racine carrée d'un nombre négatif), le symbole 1.#NAN ou -1.#NAN est affiché.

La précision de représentation est de 2-24. Pour la visualisation de nombre flottant, il est inutile d'afficher plus de 6 chiffres après la virgule.

Note :
- La valeur "1 285" est interprétée en tant que valeur entière. Pour pouvoir être prise en compte comme valeur flottante, elle doit être écrite sous la forme suivante : "1285.0"

Plage limite des fonctions arithmétiques sur des objets flottants

Le tableau suivant décrit la plage limite des fonctions arithmétiques sur des objets flottants :

Fonction arithmétique		**Plage limite et opérations invalides**	
Type	**Syntaxe**	**#QNAN (Non valide)**	**#INF (Infini)**
Racine carrée d'un opérande	SQRT(x)	x < 0	x > 1,7E38
Alimentation d'un entier par un réel EXPT(%MF,%MW)	EXPT(y, x) (où : x^y = %MW^%MF)	x < 0	y.ln(x) > 88
Logarithme de base 10	LOG(x)	x <= 0	x > 2,4E38
Logarithme naturel	LN(x)	x <= 0	x > 1,65E38
Exponentiel naturel	EXP(x)	x < 0	x > 88.0

Compatibilité matérielle

Les opérations sur flottants et mots doubles ne sont pas prises en charge par tous les automates Twido.

Le tableau suivant décrit la compatibilité matérielle :

Automate Twido	**Mots doubles pris en charge**	**Flottants pris en charge**
TWDLMDA40DUK	Oui	Oui
TWDLMDA40DTK	Oui	Oui
TWDLMDA20DUK	Oui	Non
TWDLMDA20DTK	Oui	Non
TWDLMDA20DRT	Oui	Oui
TWDLC••40DRF	Oui	Oui
TWDLC•A24DRF	Oui	Non
TWDLC•A16DRF	Oui	Non
TWDLC•A10DRF	Non	Non

Contrôle de validité

Le bit système %S18 est mis à 1 lorsque le résultat ne se situe pas dans la plage valide.

Les bits de mot d'état %SW17 indiquent la cause d'une erreur au niveau d'une opération sur valeur flottante.

Différents bits du mot %SW17 :

%SW17:X0	Opération incorrecte, le résultat n'est pas un nombre (1.#NAN ou -1.#NAN)
%SW17:X1	Réservé
%SW17:X2	Division par 0, le résultat est l'infini (-1.#INF ou 1.#INF)
%SW17:X3	Résultat supérieur en valeur absolue à +3,402824e+38, le résultat est l'infini (-1.#INF ou 1.#INF)
%SW17:X4 à X15	Réservé

Ce mot est remis à 0 par le système lors d'un démarrage à froid et par le programme pour une réutilisation.

Description des objets flottants et mots doubles

Le tableau suivant décrit les objets flottants et mots doubles :

Type d'objet	Description	Repère	Nombre maximal	Accès en écriture	Forme indexée
Valeurs immédiates	Entiers (mot double) ou décimaux (flottant) dont le format est identique à des objets de 32 bits.	-	[-]	Non	-
Flottant interne	Objets utilisés pour stocker des valeurs dans la mémoire des données lorsque le système est en cours d'exécution.	%MFi	1500	Oui (ODM/T)	%MFi[index]
Mot double interne		%MDi	1500	Oui (ODM/T)	%MDi[index]
Constante flottante	Mémorise les constantes.	%KFi	128	Oui, (T)	%KFi[index]
Constante double		%KDi	128	Oui, (T)	%KDi[index]
Remarque :	**1. ODM : Accès en écriture via le module d'affichage (reportez-vous à la rubrique *p. 433*)** **2. T : Accès en écriture via TwidoSuite**				

Possibilité de recouvrement entre objets

Les mots longueur simple, double et flottants sont stockés au sein de l'espace des données dans une même zone mémoire. Ainsi, le mot flottant %MFi et le mot double %MDi correspondent aux mots longueur simple %MWi et %MWi+1 (le mot %MWi contenant les bits de poids faible et le mot %MWi+1 les bits de poids fort du mot %MFi).

Le tableau suivant illustre le recouvrement des mots flottants et des mots doubles internes :

Flottant et double	Repère impair	Mots internes
%MF0 / %MD0		%MW0
	%MF1 / %MD1	%MW1
%MF2 / %MD2		%MW2
	%MF3 / %MD3	%MW3
%MF4 / %MD4		%MW4
	...	%MW5
...		...
	%MFi / %MDi	%MWi
%MFi+1 / %MDi+1		%MWi+1

Le tableau suivant illustre le recouvrement des constantes flottantes et doubles :

Flottant et double	Repère impair	Mots internes
%KF0 / %KD0		%KW0
	%KF1 / %KD1	%KW1
%KF2 / %KD2		%KW2
	%KF3 / %KD3	%KW3
%KF4 / %KD4		%KW4
	...	%KW5
...		...
	%kFi / %kDi	%KWi
%KFi+1 / %KDi+1		%KWi+1

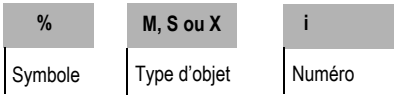
Exemple :

%MF0 correspond à %MW0 et %MW1. %KF543 correspond à %KW543 et %KW544.

Adressage d'objets bits

Syntaxe

L'adressage des objets bits d'étape, internes et système doit se conformer à la syntaxe suivante :



Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

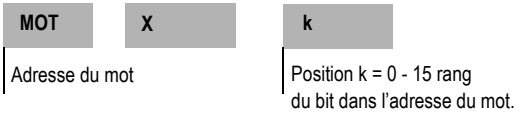
Groupe	Elément	Description
Symbole	%	Une variable logicielle doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les bits internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	S	Les bits système donnent des informations d'état et de contrôle relatives à l'automate.
	X	Les bits d'étape offrent des informations sur l'état des activités des étapes.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

Exemples d'adressage d'objets bits :

- %M25 = bit interne numéro 25
- %S20 = bit système numéro 20
- %X6 = bit étape numéro 6

Objets bits extraits de mots

TwidoSuite permet d'extraire un des 16 bits des mots. L'adresse du mot est alors complétée par le rang du bit extrait suivant la syntaxe suivante :



Exemples :

- %MW5:X6 = bit numéro 6 du mot interne %MW5
- %QW5.1:X10 = bit numéro 10 du mot de sortie %QW5.1

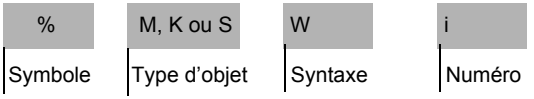
Adressage d'objets mots

Introduction

L'adressage d'objets mots doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *p. 41*) et des blocs fonction (reportez-vous à la rubrique *p. 44*).

Syntaxe

L'adressage des mots internes, constante et système doit se conformer à la syntaxe suivante :



Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Elément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les mots internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les mots constante permettent de stocker des valeurs constantes ou des messages alphanumériques. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSuite.
	S	Les mots système offrent des informations d'état et de régulation relatives à l'automate.
Syntaxe	W	Mot de 16 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

Exemples d'adressage d'objets mots :

- %MW15 = mot interne numéro 15
- %KW26 = mot constante numéro 26
- %SW30 = mot système numéro 30

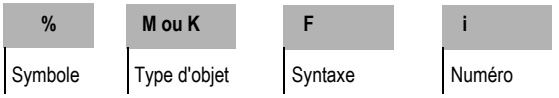
Adressage d'objets flottants

Présentation

L'adressage d'objets flottants doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *p. 41*) et des blocs fonctions (reportez-vous à la rubrique *p. 44*).

Syntaxe

L'adressage des flottants internes et constants doit se conformer à la syntaxe suivante :



Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Elément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les flottants internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les flottants constants permettent de stocker des valeurs constantes. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSuite.
Format	F	Objet de 32 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

Exemples d'adressage d'objets flottants :

- %MF15 = flottant interne numéro 15
- %KF26 = flottant constant numéro 26

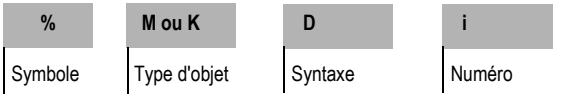
Adressage d'objets mots doubles

Présentation

L'adressage des objets mots doubles doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *p. 41*) et des blocs fonctions (reportez-vous à la rubrique *p. 44*).

Syntaxe

L'adressage des mots doubles internes et constants doit se conformer à la syntaxe suivante :



Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Elément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les mots doubles internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les mots doubles constants permettent de stocker des valeurs constantes ou des messages alphanumériques. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSuite.
Format	D	Double mot de 32 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

Exemples d'adressage d'objets mots doubles :

- %MD15 = mot double interne numéro 15
- %KD26 = mot double constant numéro 26

Repérage des entrées/sorties

Introduction

Chaque point d'E/S (entrée/sortie) d'une configuration Twido possède un repère unique. Par exemple, le repère "%I0.0.4" est affecté à l'entrée 4 d'un automate.

Des repères d'E/S peuvent être affectés aux matériels suivants :

- Automate configuré en tant que maître de liaison distante
- Automate configuré en tant qu'E/S distante
- Modules d'E/S d'expansion

Le module d'interface bus AS-Interface TWDNOI10M3 et le module bus de terrain CANopen TWDNCO1M utilisent chacun leur propre système d'adressage des entrées/sorties des équipements esclaves reliés à leur bus :

- Pour le module TWDNOI10M3, voir *p. 245*.
- pour le module TWDNCO1M, voir *p. 287*.

Références multiples à une sortie ou à une bobine

Un programme peut comporter plusieurs références à une même sortie ou bobine. Seul le résultat de la dernière référence traitée est mis à jour au niveau des sorties du matériel. Par exemple, %Q0.0.0 peut être utilisé plusieurs fois dans un programme sans qu'un avertissement ne signale la multiplicité des occurrences. Il est donc important de ne valider que l'équation qui donnera l'état souhaité de la sortie.

⚠ ATTENTION

OPÉRATION INATTENDUE

Les doublons de sortie ne sont pas contrôlés et aucun avertissement n'est donné. Vérifiez l'utilisation qui est faite des sorties et des bobines avant de les modifier dans l'application.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Format

L'adressage des entrées et des sorties doit se conformer à la syntaxe suivante.

%	I, Q	x	.	y	.	z
Symbole	**Type d'objet**	**Position de l'automate**	**point**	**Type d'E/S**	**point**	**Numéro de voie**

Le repérage des mots d'échange en entrée et en sortie doit se conformer à la syntaxe ci-dessous.

%	I, Q	W	x	.	y
Symbole	**Type d'objet**	**Format**	**Position de l'automate**	**point**	**Type d'E/S**

Description

Le tableau suivant décrit la syntaxe de repérage des E/S.

Groupe	Elément	Valeur	Description
Symbole	%	-	Un repère interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	I	-	Entrée. "Image logique" de l'état électrique de l'entrée d'un automate ou d'un module d'E/S d'expansion.
	Q	-	Sortie. "Image logique" de l'état électrique de la sortie d'un automate ou d'un module d'E/S d'expansion.
Position de l'automate	x	0 1 - 7	Automate maître (maître de liaison distante). Automate distant (esclave de liaison distante).
Type d'E/S	y	0 1 - 7	E/S de base (E/S locale sur un automate). Modules d'E/S d'expansion.
Numéro de voie	z	0 - 31	Numéro de la voie d'E/S sur l'automate ou le module d'E/S d'expansion. Le nombre de points d'E/S disponibles dépend du modèle de l'automate ou du type du module d'E/S d'expansion.

Exemples

Le tableau suivant présente quelques exemples de repérage des E/S.

Objet d'E/S	Description
%I0.0.5	Entrée n° 5 sur la base automate (E/S locale).
%Q0.3.4	Sortie n° 4 sur le module d'E/S d'expansion d'adresse 3 pour la base automate (E/S d'expansion).
%I0.0.3	Entrée n° 3 sur la base automate.
%I3.0.1	Entrée n° 1 sur l'automate d'E/S distant d'adresse 3 de la liaison distante.
%I0.3.2	Entrée n° 2 sur le module d'E/S d'expansion d'adresse 3 pour la base automate.

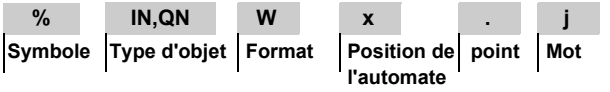
Network Addressing

Introduction

Les mots réseau %INW et %QNW permettent d'échanger des données d'application entre les automates d'extension et l'automate maître sur un réseau de liaison distante Twido. Reportez-vous au chapitre *p. 73* pour obtenir plus d'informations.

Format

L'adressage réseau doit se conformer à la syntaxe suivante.



Description de la syntaxe

Le tableau suivant décrit la syntaxe d'adressage réseau.

Groupe	Elément	Valeur	Description
Symbole	%	-	Un repère interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	IN	-	Mot d'entrée réseau. Transfert de données de l'automate maître vers l'automate d'extension.
	QN	-	Mot de sortie réseau. Transfert de données de l'automate d'extension vers l'automate maître.
Format	W	-	Mot de 16 bit.
Position de l'automate	x	0 1 - 7	Automate maître (maître de liaison distante). Automate distant (esclave de liaison distante).
Mot	j	0 - 3	Chaque automate d'extension utilise un maximum de quatre mots pour assurer l'échange de données avec l'automate maître.

Exemples

Le tableau suivant présente quelques exemples d'adressage réseau.

Objet réseau	Description
%INW3.1	Mot réseau n°1 de l'automate distant n°3.
%QNW0.3	Mot réseau n°3 de la base automate.

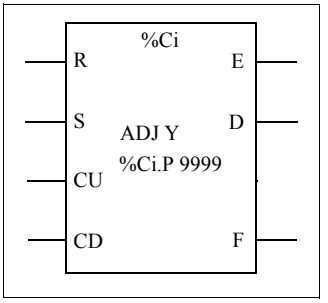
Objets blocs fonction

Introduction

Les blocs fonction contiennent des objets bits et des mots spécifiques accessibles par le programme.

Exemple de bloc fonction

L'illustration suivante présente un bloc fonction compteur.



Bloc compteur/décompteur

Objets bits

Les objets bits correspondent aux sorties des blocs. Les instructions booléennes de test peuvent accéder à ces bits selon l'une ou l'autre de ces méthodes :

- directement (LD E, par exemple) si les bits sont reliés au bloc par une programmation réversible (voir *p. 453*) ;
- en spécifiant le type de bloc (LD %Ci.E, par exemple).

Les entrées sont accessibles sous forme d'instructions.

Objets mots

Les objets mots correspondent aux paramètres et valeurs spécifiés suivants :

- Paramètres de configuration des blocs : le programme peut accéder à certains paramètres (paramètres de présélection, par exemple), mais pas à d'autres (base temps, par exemple).
- Valeurs courantes : %Ci.V, la valeur de comptage courante, par exemple.

Objets mots doubles

Les objets mots doubles augmentent les capacités de calcul de votre automate Twido lors de l'exécution de fonctions système telles que les compteurs rapides (%FC ou %VFC) et les générateurs d'impulsions (%PLS).

Le repérage des objets mots doubles 32 bits utilisés avec les blocs fonction consiste uniquement à ajouter des objets mots standard avec le caractère "D" à la syntaxe d'origine. L'exemple suivant indique comment repérer la valeur courante d'un compteur rapide (FC) au format standard et au format mot double.

- %FCi.V est la valeur courante du compteur rapide (FC) au format standard.
- %FCi.VD est la valeur courante du compteur rapide (FC) au format mot double.

Note : Les objets mots doubles ne sont pas pris en charge par tous les automates Twido. Reportez-vous au sous-chapitre *p. 34* pour savoir si votre automate Twido accepte les mots doubles.

Objets accessibles par le programme

Reportez-vous aux sous-chapitres suivants pour connaître la liste des objets accessibles par le programme.

- Pour les blocs fonction élémentaires, reportez-vous au sous-chapitre *p. 451*.
- Pour les blocs fonction avancés, reportez-vous au sous-chapitre *p. 504*.

Objets structurés

Introduction

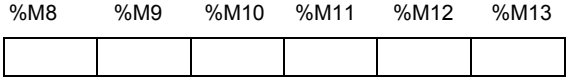
Les objets structurés sont des ensembles formés par des objets adjacents. Twido prend en charge les types d'objet structuré suivants :

- Chaînes de bits
- Tables de mots
- Tables de mots doubles
- Tables de mots flottants

Chaînes de bits

Les chaînes de bits sont composées d'une série de bits objet adjacent du même type et dont la longueur (L) est définie.

Exemple : Chaîne de bits %M8:6



Note : %M8:6 est acceptable (car 8 est un multiple de 8), alors que %M10:16 ne l'est pas (10 n'est pas un multiple de 8).

Les chaînes de bits peuvent être utilisées avec l'instruction d'affectation (voir *p. 476*).

Types de bit disponibles

Types de bit disponibles pour les chaînes de bits :

Type	Repère	Taille maximale	Accès en écriture
Bits d'entrée TOR	%I0.0:L ou %I1.0:L (1)	0<L<17	Non
Bits de sortie TOR	%Q0.0:L ou %Q1.0:L (1)	0<L<17	Oui
Bits système	%Si:L où "i" est multiple de 8	0<L<17 et i+L≤ 128	En fonction de i
Bits pas Grafcet	%Xi:L où "i" est multiple de 8	0<L<17 et i+L≤ 95 (2)	Oui (via le programme)
Bits internes	%Mi:L où "i" est multiple de 8	0<L<17 et i+L≤ 256 (3)	Oui

Légende :

1. Seuls les bits d'E/S 0 à 16 peuvent être lus en chaîne de bits. Pour les automates à 24 entrées et les modules à 32 E/S, les bits supérieurs à 16 ne peuvent pas être lus en chaîne de bits.

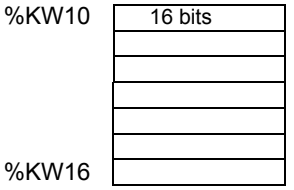
2. Le maximum de i+L pour les automates TWWDLCAA10DRF et TWDLCAA16DRF est 62.

3. Le maximum de i+L pour les automates TWWDLCAA10DRF et TWDLCAA16DRF est 128.

Tables de mots

Les tables de mots sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.

Exemple : Table de mots %KW10:7



Les tables de mots peuvent être utilisées avec l'instruction d'affectation (voir *p. 476*).

Types de mot disponibles

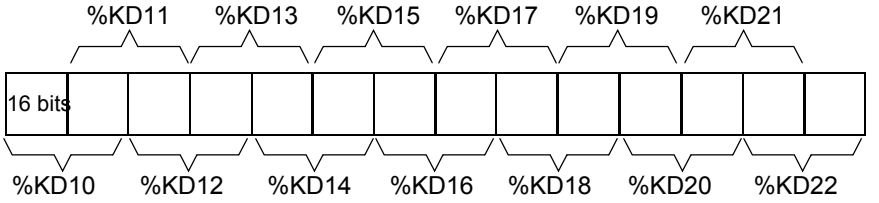
Types de mot disponibles pour les tables de mots :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MWi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KWi:L	0<L<256 et i+L< 256	Non
Mots système	%SWi:L	0<L et i+L<128	En fonction de i

Tables de mots doubles

Les tables de mots doubles sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.

Exemple : Table de mots doubles %KD10:7



Les tables de mots doubles peuvent être utilisées avec l'instruction d'affectation (voir *p. 476*).

Types de mot double disponibles

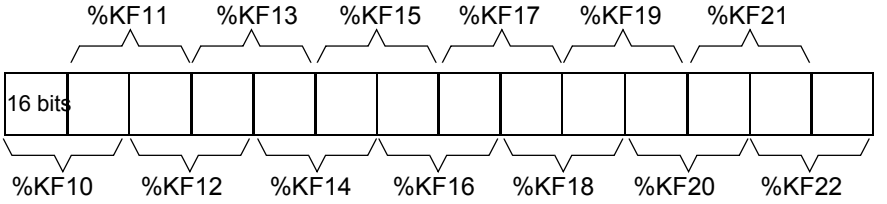
Types de mot disponibles pour les tables de mots doubles :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MDi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KDi:L	0<L et i+L< 256	Non

Tables de mots flottants

Les tables de mots flottants sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.

Exemple : Table de mots flottants %KF10:7



Les tables de mots flottants peuvent être utilisées avec l'instruction d'affectation (voir Instructions d'affectation).

Types de mot flottant disponibles

Types de mot disponibles pour les tables de mots flottants :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MFi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KFi:L	0<L et i+L<256	Non

Objets indexés

Présentation

Un mot indexé est un mot simple ou double ou un flottant comportant un repère d'objet indexé. Il existe deux types de repérage d'objet :

- repérage direct
- repérage indexé

Repérage direct

Le repère direct d'un objet est défini au moment de l'écriture du programme.

Exemple : %M26 est un bit interne dont le repère direct est 26.

Repérage indexé

L'indexation du repère d'un objet permet de modifier ce repère en attribuant un index au repère direct d'un objet. Le contenu de l'index est ajouté au repère direct de l'objet. L'index est défini par un mot interne %MWi. Le nombre de "mots indexés" est illimité.

Exemple : %MW108[%MW2] est un mot dont le repère est composé du repère direct 108 et du contenu du mot %MW2.

Si la valeur du mot %MW2 est 12, le fait d'écrire dans %MW108[%MW2] équivaut à écrire dans %MW120 (108 + 12).

Objets disponibles pour le repérage indexé

Le tableau suivant répertorie les différents types d'objet disponibles pour le repérage indexé.

Type	Adresse	Taille maximale	Accès en écriture
Mots internes	%MWi[MWj]	0≤ i+%MWj<3000	Oui
Mots constante	%KWi[%MWj]	0≤ i+%MWj<256	Non
Mots doubles internes	%MDi[MWj]	0≤ i+%MWj<2999	Oui
Mots doubles constante	%KDi[%MWj]	0≤ i+%MWj<255	Non
Flottants internes	%MFi[MWj]	0≤ i+%MWj<2999	Oui
Flottants constante	%KFi[%MWj]	0≤ i+%MWj<255	Non

Les objets indexés peuvent être utilisés avec les instructions d'affectation (voir *p. 559* pour mots simples et doubles) et dans les instructions de comparaison (voir *p. 564* pour mots simples et doubles). Ce type de repérage permet de scruter individuellement un ensemble d'objets du même type (tels que des mots internes ou des constantes), en modifiant le contenu de l'objet indexé via le programme.

Bit système de débordement d'index %S20

Un débordement d'index se produit lorsque le repère d'un objet indexé dépasse les limites de la zone mémoire contenant le même type d'objet. Pour résumer :

- Le repère de l'objet plus le contenu de l'index sont inférieurs à 0.
- Le repère de l'objet plus le contenu de l'index sont supérieurs au plus grand mot directement référencé dans l'application. Le nombre maximum est 2 999 (pour les mots %MWi) ou 255 (pour les mots %KWi).

En cas de débordement d'index, le système provoque la mise à 1 du bit système %S20 et une valeur d'index égale à 0 est affectée à l'objet.

Note : L'utilisateur est responsable du contrôle des débordements. Le bit %S20 doit être lu par le programme utilisateur pour un traitement éventuel. La remise à zéro est à la charge de l'utilisateur.
%S20 (état initial = 0) :

- Sur débordement d'index : mise à 1 par le système.
- Acquittement de débordement : mise à 0 par l'utilisateur, après modification de l'index.

Symbolisation d'objets

Introduction

Les symboles permettent de répérer des objets du langage logiciel Twido, à l'aide de noms ou de mnémoniques personnalisés. L'utilisation de symboles permet de procéder à une analyse et à un examen rapides de la logique du programme, et simplifie considérablement les phases de développement et de test d'une application.

Exemple

Par exemple, WASH_END est un symbole pouvant être utilisé pour identifier un bloc fonction de type temporisateur représentant la fin d'un cycle de lavage. L'utilisation de ce nom se révélera beaucoup plus pratique que celui du repère du programme, tel que %TM3.

Instructions pour la définition de symboles

Les noms de symboles doivent répondre aux exigences suivantes :

- 32 caractères au maximum.
- des lettres (A-Z), chiffres (0-9) ou traits de soulignement (_).
- une lettre ou un caractère accentué comme premier caractère. Ces noms ne peuvent pas comporter de signe de pourcentage (%).
- Ces noms ne peuvent pas contenir d'espaces ou de caractères spéciaux.
- des majuscules ou des minuscules. Par exemple, Pump1 et PUMP1 représentent le même symbole et ne peuvent être utilisés qu'une seule fois dans une application.

Edition des symboles

Utilisez l'éditeur de symboles pour définir et associer des objets de langage. Il est important de signaler que les symboles et leurs commentaires ne sont pas stockés sur l'automate, mais avec l'application, sur le disque dur. Il est donc impossible de transférer ces symboles vers l'automate, avec l'application.

Mémoire utilisateur

3

Présentation

Objet de ce chapitre

Ce chapitre offre une description de la structure de la mémoire utilisateur Twido, ainsi que des informations sur son utilisation.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Structure de la mémoire utilisateur	54
Backup et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue	58
Backup et restauration avec une cartouche de sauvegarde de 32 Ko	60
Utilisation de la cartouche de mémoire étendue 64 Ko	63

Structure de la mémoire utilisateur

Introduction

La mémoire de l'automate accessible par votre application est divisée en deux ensembles distincts :

- les valeurs de bits ;
- les valeurs de mots (valeurs signées à 16 bits) et les valeurs de mots doubles (valeurs signées à 32 bits).

Mémoire bits

La mémoire bits est située dans la mémoire RAM intégrée de l'automate. Elle contient l'image des 128 objets bits.

Mémoire mots

La mémoire mots (16 bits) prend en charge les éléments suivants :

- **Mots dynamiques** : mémoire d'exécution (stockée uniquement dans la RAM).
- **Mots (%MW) et mots doubles (%MD) mémoire** : données dynamiques système et données système.
- **Programme** : descripteurs et code exécutable des tâches.
- **Données de configuration** : mots constante, valeurs initiales et configuration des entrées/sorties.

Types de stockage mémoire

Les automates Twido disposent des trois types de stockage mémoire suivant :

- RAM
  Mémoire volatile interne : contient des mots dynamiques, des mots mémoire, des données de configuration et de programme.
- EEPROM
  Mémoire EEPROM intégrée de 32 Ko permettant une sauvegarde interne des données et du programme. Elle protège le programme des altérations causées par une défaillance de pile ou une coupure secteur de plus de 30 jours. Elle contient des données de programme et de configuration. Elle comporte un maximum de 512 mots mémoire. Le programme n'est pas sauvegardé si une cartouche de mémoire étendue de 64 Ko est en cours d'utilisation et que Twido a été configuré pour accepter cette cartouche de mémoire. Twido Extreme TWDLEDCK1 ne possède pas de cartouche de mémoire étendue.
- Cartouche de sauvegarde de 32 Ko
  Cartouche externe en option utilisée pour enregistrer un programme et transférer ce programme vers d'autres automates Twido. Elle peut être utilisée pour mettre à jour le programme dans la RAM de l'automate. Elle contient un programme et des constantes, mais aucun mot mémoire. Twido Extreme TWDLEDCK1 ne possède aucune cartouche de sauvegarde.
- Cartouche de mémoire étendue de 64 Ko
  Cartouche externe en option qui stocke un programme jusqu'à 64 Ko. Doit rester raccordée à l'automate tant que le programme est utilisé. Twido Extreme TWDLEDCK1 ne possède pas de cartouche de mémoire étendue.

Enregistrement de la mémoire

Les mots mémoire et le programme de votre automate peuvent être enregistrés dans les éléments suivants :

- RAM (jusqu'à 30 jours avec une pile satisfaisante)
- EEPROM (32 Ko maximum)

Le transfert du programme depuis la mémoire EEPROM vers la mémoire RAM s'effectue automatiquement, lorsqu'il y a perte du programme dans la RAM (ou en cas d'absence de pile). Twido Extreme TWDLEDCK1 ne possède aucune pile interne.

Notez qu'il est également possible d'effectuer un transfert manuel à l'aide de TwidoSuite.

Configurations de la mémoire pour les automates compacts ou modulaires Twido

Les tableaux suivants présentent les configurations de mémoire possibles des automates Twido (compacts et modulaires).

Type de mémoire	Automates compacts				
	10DRF	**16DRF**	**24DRF**	**40DRF (32 Ko)**	**40DRF** (64 Ko)**
RAM interne Mém 1*	10 Ko	10 Ko	10 Ko	10 Ko	10 Ko
RAM externe Mém 2*		16 Ko	32 Ko	32 Ko	64 Ko
EEPROM interne	8 Ko	16 Ko	32 Ko	32 Ko	32 Ko***
EEPROM externe	32 Ko	32 Ko	32 Ko	32 Ko	64 Ko
Taille maximale du programme	8 Ko	16 Ko	32 Ko	32 Ko	64 Ko
Sauvegarde externe maximale	8 Ko	16 Ko	32 Ko	32 Ko	64 Ko

Type de mémoire	Automates modulaires		
	20DUK 20DTK	**20DRT 40DUK 40DTK (32 Ko)**	**20DRT 40DUK 40DTK** (64 Ko)**
RAM interne Mém 1*	10 Ko	10 Ko	10 Ko
RAM externe Mém 2*	32 Ko	32 Ko	64 Ko
EEPROM interne	32 Ko	32 Ko	32 Ko***
EEPROM externe	32 Ko	32 Ko	64 Ko
Taille maximale du programme	32 Ko	32 Ko	64 Ko
Sauvegarde externe maximale	32 Ko	32 Ko	64 Ko

(*) Mém 1 et Mém 2 en utilisation mémoire.

(**) dans ce cas la cartouche 64 Ko doit être installée sur le Twido et déclarée dans la configuration, si elle n'est pas déjà déclarée,

(***) réservé à la sauvegarde des premiers 512 mots %MW ou des premiers 256 mots doubles %MD.

Configurations de la mémoire Twido Extreme

Le tableau suivant présente les types de configuration de mémoire possibles pour les automates Twido Extreme.

Type de mémoire	Automate Twido Extreme
	TWDLEDCK1
RAM externe	32 Ko
EEPROM interne	32 Ko
Taille maximale du programme	32 Ko

Backup et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue

Introduction

Ce sous-chapitre détaille les fonctions de backup et de restauration de la mémoire dans les automates modulaires et compacts sans cartouche backup, ni cartouche de mémoire étendue connectée.

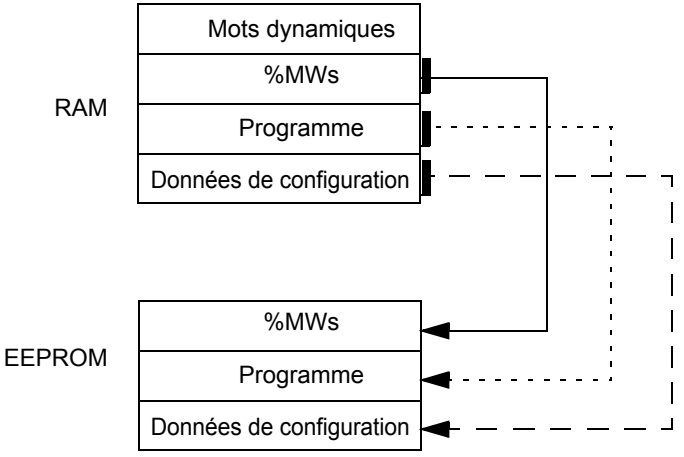
Cette section ne s'applique pas à l'automate TWDLEDCK1 Twido Extreme, qui ne possède pas de cartouche de sauvegarde. D'autres informations sont données à la section concernant la mémoire utilisateur de Twido Extreme.

Présentation

Les programmes Twido, les mots mémoire et les données de configuration peuvent être sauvegardés à l'aide de l'EEPROM interne des automates. Etant donné que le backup d'un programme dans l'EEPROM interne efface tout mot mémoire préalablement sauvegardé, effectuez tout d'abord le backup du programme, puis des mots mémoire configurés. Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. Si aucun programme n'est sauvegardé dans l'EEPROM interne, vous ne pouvez pas y sauvegarder des mots mémoire.

Structure de la mémoire

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate. Les flèches montrent les éléments pouvant être sauvegardés dans l'EEPROM depuis la RAM :



Backup du programme

Pour sauvegarder votre programme dans l'EEPROM, procédez comme suit.

Etape	Action
1	L'élément suivant doit être vérifié : Le programme dans la RAM est valide.
2	Dans la fenêtre TwidoSuite, déroulez le menu Automate et cliquez sur Backup.

Restauration du programme

Lors de la mise sous tension, il existe une méthode pour restaurer le programme dans la RAM depuis l'EEPROM (si aucune cartouche ou mémoire étendue n'est en place) :

- Le programme de la RAM n'est pas valide

Pour restaurer un programme manuellement depuis l'EEPROM, procédez comme suit :

- Dans la fenêtre TwidoSuite, déroulez le menu Automate et cliquez sur Restauration.

Backup des données (%MWs)

Pour effectuer le backup de vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vérifiés : Un programme valide est présent dans la RAM (%SW96:X6=1). Le même programme valide est déjà sauvegardé dans l'EEPROM. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. **Remarque :** La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

Restauration des données (%MWs)

Restaurez %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vérifiés :

- Une application backup valide est présente dans l'EEPROM.
- L'application dans la RAM correspond à l'application backup dans l'EEPROM.
- Les mots mémoire backup sont valides.

Backup et restauration avec une cartouche de sauvegarde de 32 Ko

Introduction

Ce sous-chapitre détaille les fonctions de backup et de restauration de la mémoire des automates modulaires et compacts équipés d'une cartouche de sauvegarde de 32 Ko.

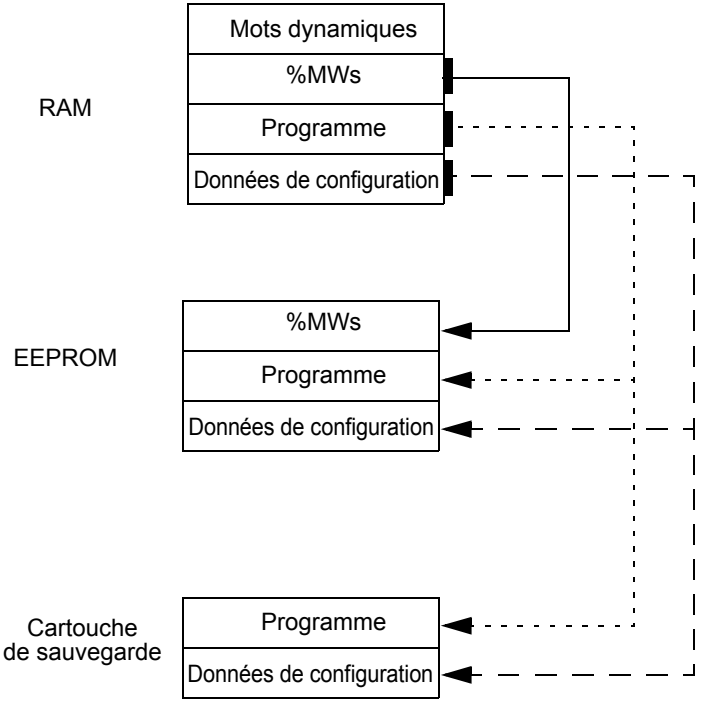
Cette section ne s'applique pas à l'automate TWDLEDCK1 Twido Extreme, qui ne possède pas de cartouche de sauvegarde. D'autres informations sont données à la section sur la mémoire utilisateur de Twido Extreme.

Présentation

La cartouche de backup est utilisée pour sauvegarder un programme et le transférer vers d'autres automates Twido. Elle doit être retirée d'un automate et mise de côté une fois le programme installé ou sauvegardé. Seules les données du programme et les données de configuration peuvent être sauvegardées dans la cartouche (%MWs ne peut pas être sauvegardé dans la cartouche de backup de 32 Ko). Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. Une fois l'installation du programme terminée, tout %MWs sauvegardé dans l'EEPROM interne avant l'installation sera perdu.

Structure de la mémoire

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate avec une cartouche de backup connectée. Les flèches montrent les éléments pouvant être sauvegardés dans l'EEPROM et la cartouche depuis la RAM :



Backup du programme

Pour effectuer un backup de votre programme dans la cartouche de backup, procédez comme suit :

Etape	Action
1	Mettez l'automate hors tension.
2	Raccordez la cartouche de backup.
3	Mettez l'automate sous tension.
4	Dans la fenêtre TwidoSuite, déroulez le menu Automate et cliquez sur Backup.
5	Mettez l'automate hors tension.
6	Retirez la cartouche de backup de l'automate.

Restauration du programme

Pour charger un programme sauvegardé sur une cartouche de backup dans un automate, procédez comme suit :

Etape	Action
1	Mettez l'automate hors tension.
2	Raccordez la cartouche de backup.
3	Mettez l'automate sous tension. (Si le démarrage automatique est configuré, vous devez à nouveau effectuer la mise sous tension pour entrer en mode d'exécution.)
4	Mettez l'automate hors tension.
5	Retirez la cartouche de backup de l'automate.

Backup des données (%MWs)

Pour effectuer le backup de vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vérifiés : Un programme valide est présent dans la RAM. Le même programme valide est déjà sauvegardé dans l'EEPROM. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. **Remarque :** La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

Restauration des données (%MWs)

Restaurez %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vérifiés :

- Une application backup valide est présente dans l'EEPROM.
- L'application dans la RAM correspond à l'application backup dans l'EEPROM.
- Les mots mémoire backup sont valides.

Utilisation de la cartouche de mémoire étendue 64 Ko

Introduction

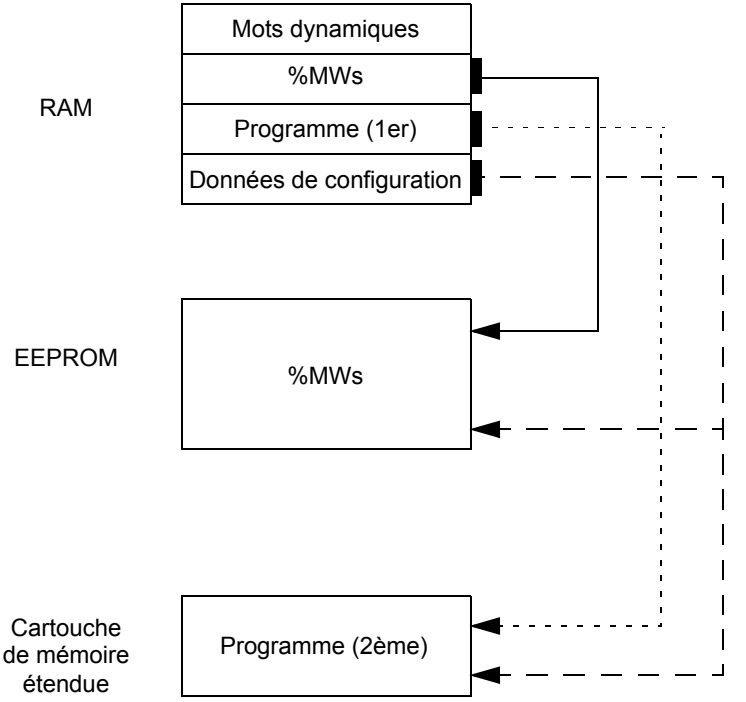
Ce sous-chapitre détaille l'utilisation des fonctions de mémoire dans les automates modulaires équipés d'une cartouche de mémoire étendue 64 Ko.

Présentation

La cartouche de mémoire étendue 64 Ko est utilisée pour étendre la capacité de mémoire du programme de votre automate Twido de 32 à 64 Ko. Elle doit rester raccordée à l'automate tant que le programme étendu est utilisé. Si la cartouche est retirée, l'automate s'arrête. Le backup des mots mémoire est quand même effectué dans l'EEPROM de l'automate. Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. La cartouche de mémoire étendue 64 Ko présente le même comportement à la mise sous tension que la cartouche de backup 32 Ko.

Structure de la mémoire

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate utilisant une cartouche de mémoire étendue. Les flèches indiquent les éléments sauvegardés dans l'EEPROM et la cartouche de mémoire étendue 64 Ko depuis la RAM :



Configuration du logiciel et installation de la mémoire étendue

Avant de procéder à l'écriture de votre programme étendu, vous devez installer la cartouche de mémoire étendue 64 Ko dans votre automate. Voici les quatre étapes à suivre :

Etape	Action
1	Sous l'option Matériel de la fenêtre de votre logiciel Twido, saisissez "TWDXCPMFK64".
2	Mettez l'automate hors tension.
3	Raccordez la cartouche de mémoire étendue 64 Ko.
4	Mettez l'automate sous tension.

Enregistrez votre programme.

Une fois votre cartouche de mémoire étendue 64 Ko installée et votre programme écrit :

- Dans la fenêtre du logiciel Twido, déroulez le menu Automate et cliquez sur Backup.

Données (%MWs) Backup

Pour effectuer le backup de vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vérifiés : Un programme valide est présent. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. **Remarque :** La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

Restauration des données (%MWs)

Restaurez %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vérifiés :

- Un programme valide est présent.
- Les mots mémoire backup sont valides.

Gestion des tâches événementielles

4

Présentation

Objet de ce chapitre

Ce chapitre décrit les tâches événementielles et leur exécution dans l'automate.

Note : Les tâches événementielles ne sont pas gérées par l'automate Twido Brick 10 (TWDLCAA10DRF).

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble des tâches événementielles	66
Description des différentes sources d'événement	67
Gestion des événements	69

Vue d'ensemble des tâches événementielles

Présentation

Le précédent chapitre présente les tâches périodiques (Scrutation périodique, Twido, Guide de référence du matériel, Bases compactes et modulaires) et cycliques (Scrutation cyclique, Twido, Guide de référence du matériel, Bases compactes et modulaires) où les objets sont mis à jour en début et fin de tâche. Des sources d'événements peuvent provoquer des interruptions de cette tâche pendant lesquelles des tâches plus prioritaires (événementielles) sont exécutées pour permettre une mise à jour plus rapide des objets.

Une tâche événementielle :

- est une portion de programme exécutée à une condition donnée (source d'événement),
- possède une priorité plus haute que le programme principal,
- garantit un temps de réponse rapide qui permet de réduire le temps de réponse du système.

Description d'un événement

Un événement se compose :

- d'une source d'événement qui peut être défini comme une condition d'interruption logicielle ou matérielle qui interrompt le programme principal (voir *p. 67*),
- d'une section qui est une entité autonome de programmation liée à un événement ;
- d'une file d'événements permettant de stocker la liste des événements jusqu'à leur exécution ;
- d'une priorité qui est l'ordre d'exécution de l'événement.

Description des différentes sources d'événement

Vue d'ensemble des différentes sources d'événement

Une source d'événement nécessite d'être gérée par le logiciel pour assurer l'interruption du programme principal par l'événement et l'appel de la section de programmation liée à l'événement. Le temps de scrutation de l'application n'a pas d'effet sur l'exécution des événements.

Les 9 sources d'événements permises sont les suivantes :

- 4 conditions liées aux seuils des blocs fonction compteur rapide (2 événements par instance de %VFC),
- 4 conditions liées aux entrées physiques d'une base automate,
- 1 condition périodique.

Une source d'événement ne peut être attachée qu'à un seul événement et doit être immédiatement détectée par TwidoSuite. Sitôt détectée, le logiciel exécute la section de programmation attachée à l'événement : chaque événement est attaché à un sous-programme portant une étiquette **SRi:** définie lors de la configuration des sources d'événement.

Evénement sur entrées physiques d'une base automate

Les entrées %I0.2, %I0.3, %I0.4 et %I0.5 peuvent être utilisées comme sources d'événement, à condition qu'elles ne soient pas verrouillées et que les événements y soient permis pendant la configuration.

Les traitements événementiels peuvent être déclenchés par les entrées 2 à 5 d'une base automate (position 0), sur front montant ou descendant.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration matérielle -> Configuration des entrées" dans l'aide en ligne du "Guide d'exploitation TwidoSuite".

Evénement sur les sorties d'un bloc fonction %VFC

Les sorties TH0 et TH1 du bloc fonction %VFC sont des sources d'événements. Les sorties TH0 et TH1 passent respectivement :

- à 1 quand la valeur est supérieure au seuil S0 et au seuil S1,
- à 0 quand la valeur est inférieure au seuil S0 et au seuil S1.

Un front montant ou descendant de ces sorties peut déclencher un traitement événementiel.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration logicielle -> Compteurs rapides" dans l'aide en ligne du "Guide d'exploitation TwidoSuite".

Evénement périodique

Cet événement exécute une même section de programmation de façon périodique. Cette tâche est plus prioritaire que la tâche principale (maître).

Cette source d'événement est moins prioritaire par contre que les autres sources d'événement.

La période de cette tâche est fixée en configuration, de 5 à 255 ms. Un seul événement périodique peut être utilisé.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration des paramètres du programme -> Mode de scrutation" dans l'aide en ligne du "Guide d'exploitation TwidoSuite".

Gestion des événements

File d'événements et priorité

Les événements présentent 2 priorités possibles : Haute ou Basse. Mais un **seul** type d'événement (donc une seule source d'événement) peut avoir la priorité Haute. Les autres événements ont alors une priorité Basse, et leur ordre d'exécution dépend alors de leur ordre de détection.

Pour gérer l'ordre d'exécution des tâches événementielles, il existe deux files d'événements :

- l'une permettant de stocker jusqu'à 16 événements de priorité Haute (d'une même source d'événement),
- l'autre permettant de stocker jusqu'à 16 événements de priorité Basse (des autres sources d'événement).

Ces files sont gérées comme des FIFO (First In First Out) : le premier événement stocké est le premier exécuté. Mais elles ne peuvent stocker que 16 événements, les événements supplémentaires sont perdus.

La file de priorité Basse n'est exécutée que lorsque la file de priorité Haute est vide.

Gestion des files d'événements

A chaque fois qu'une interruption apparaît (liée à une source d'événement), la séquence suivante est lancée :

Etape	Description
1	Gestion de l'interruption : ● connaissance de l'interruption physique, ● événement stocké dans la file d'événements appropriée, ● vérification qu'un événement de même priorité n'est pas en cours (sinon l'événement reste en attente dans sa file).
2	Sauvegarde du contexte.
3	Exécution de la section de programmation (sous-programme étiqueté SRi:) liée à l'événement.
4	Mise à jour des sorties
5	Restauration du contexte

Avant que le contexte ne soit rétabli, tous les événements de la file doivent être exécutés.

Contrôle des événements

Des bits et mots systèmes sont utilisés pour contrôler les événements (voir *p. 673*) :

- %S31 : permet d'exécuter ou de retarder un événement,
- %S38 : permet de placer ou non un événement dans la file d'événements,
- %S39 : permet de savoir si des événements sont perdus,
- %SW48 : affiche le nombre d'événements exécutés depuis le dernier démarrage à froid (compte tous les événements à l'exception des événements périodiques.)

La valeur du bit %S39 et du mot %SW48 est initialisée à zéro et celle du %S31 et du %S38 est réglée sur son état initial 1 lors d'un redémarrage à froid ou après chargement d'une application, mais reste inchangée lors d'un redémarrage à chaud. Dans tous les cas, la file d'événements est initialisée.

Fonctions spéciales

II

Présentation

Objet de cette partie

Cette rubrique décrit les communications, les fonctions analogiques intégrées, la gestion des modules d'E/S analogiques, la mise en œuvre du bus AS-Interface V2 et du bus de terrain CANopen des automates Twido.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
5	Communications	73
6	Fonctions analogiques intégrées	149
7	Gestion des modules analogiques	153
8	Configuration d'entrée/sortie Twido Extreme	167
9	Mise en œuvre du bus AS-Interface V2	219
10	Installation et configuration du bus de terrain CANopen	251
11	Installation et configuration du bus de terrain CANJ1939	297
12	Configuration de la passerelle Ethernet TwidoPort	337
13	Fonctionnement de l'afficheur	367

Communications

5

Présentation

Objet de ce chapitre

Ce chapitre offre une présentation des procédures de configuration, de programmation et de gestion des différents types de communications à l'aide d'automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation des différents types de communication	74
Communication entre TwidoSuite et l'automate	76
Communication entre TwidoSuite et un modem	83
Communications par liaison distante	96
Communications ASCII	110
Communications Modbus	123
Requêtes Modbus standard	142
Classe d'implémentation Transparent Ready (Twido série A05, Ethernet A15)	148

Présentation des différents types de communication

Présentation

Twido dispose d'un ou deux ports série de communication utilisés pour communiquer avec les automates E/S distants, les automates d'extension ou divers équipements. Les deux ports, lorsqu'ils sont disponibles, peuvent être utilisés pour tous les services, à l'exception de la communication avec TwidoSuite, qui ne peut se faire qu'avec le premier port. Trois protocoles de base sont pris en charge sur chaque automate Twido : liaison distante, ASCII ou Modbus (maître ou esclave Modbus).

En outre, les automates compacts TWDLC•E40DRF disposent d'un port de communication Ethernet RJ45. Ils prennent en charge le protocole client/serveur Modbus TCP/IP pour les communications poste à poste entre les automates sur le réseau Ethernet.

L'automate TWDLEDCK1 Twido Extreme ne possède qu'un seul port série et prend en charge les communications Modbus, ASCII, Ethernet et BlueTooth.

Liaison distante

La liaison distante est un bus maître/esclave très rapide conçu pour communiquer une petite quantité de données entre l'automate maître et un maximum de sept automates distants (esclave). Les données de l'application ou les données d'E/S sont transférées en fonction de la configuration des automates distants. Il est possible d'associer différents types d'automates, tels que des automates d'E/S distantes et des automates d'extension.

La liaison distante n'est pas prise en charge par l'automate TWDLEDCK1 Twido Extreme (ni la liaison RS485 série, ni la liaison CANJ1939).

ASCII

Le protocole ASCII est un protocole half-duplex en mode caractère simple utilisé pour transmettre et/ou recevoir une chaîne de caractères de ou vers un périphérique (imprimante ou terminal). Ce protocole est uniquement pris en charge via l'instruction "EXCH".

Pour l'automate TWDLEDCK1 Twido Extreme, si le protocole ASCII est utilisé, une tension de 0 V doit être appliquée au contact de la sangle de communication (broche 22) pour permettre la communication.

Modbus

Le protocole Modbus est un protocole maître/esclave qui permet à un maître et un seul d'obtenir des réponses provenant des esclaves ou d'agir sur requête. Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les réponses aux requêtes de diffusion générale du maître ne sont pas renvoyées.

Maître Modbus - Le mode maître Modbus permet à l'automate Twido d'envoyer à un esclave une requête Modbus et d'attendre la réponse de la part de cet esclave. Le mode maître Modbus est uniquement pris en charge via l'instruction "EXCH". Il gère les modes ASCII et RTU Modbus.

Esclave Modbus - Le mode esclave Modbus permet à l'automate Twido de répondre aux requêtes Modbus d'un maître Modbus. Il s'agit du mode de communication par défaut si aucun autre type de communication n'a été configuré. L'automate Twido prend en charge les données Modbus standard, les fonctions de contrôle et les extensions de service pour l'accès aux objets. Les modes ASCII et RTU Modbus sont pris en charge en mode esclave Modbus.

Note : 32 équipements (sans répéteurs) peuvent être installés sur un réseau RS485 (1 maître et jusqu'à 31 esclaves). Les repères correspondants peuvent être compris entre 1 et 247.

Modbus TCP/IP

Note : Le protocole Modbus TCP/IP est uniquement pris en charge par les automates compacts de la série TWDLC•E40DRF disposant d'une interface réseau Ethernet intégrée.

Les informations suivantes décrivent le protocole d'application Modbus (MBAP).

Le protocole d'application Modbus (MBAP) est un protocole de sept couches permettant une communication poste à poste entre les automates programmables industriels (API) et d'autres nœuds sur le réseau.

La mise en œuvre actuelle de l'automate Twido TWDLC•E40DRF utilise le protocole d'application Modbus via TCP/IP sur le réseau Ethernet. Les transactions du protocole Modbus sont des messages de type requête-réponse. Un automate peut être client ou serveur selon qu'il envoie des requêtes ou qu'il reçoit des réponses.

Communication entre TwidoSuite et l'automate

Présentation

Chaque automate Twido comporte un port terminal EIA RS485 intégré, avec son propre bloc d'alimentation, sur le port 1 (l'automate Twido Extreme TWDLEDCK1 ne possède qu'un seul port série). Le port 1 doit être utilisé pour la communication avec le logiciel de programmation TwidoSuite.

Aucune cartouche ou aucun module de communication en option ne peut utiliser ce port. Ce dernier est néanmoins utilisable par un modem.

Vous pouvez connecter le PC au port 1 RS485 de l'automate Twido de plusieurs façons :

- via un câble TSX PCX ;
- via une ligne téléphonique : par connexion modem ;
- via Bluetooth pour l'automate Twido Extreme TWDLEDCK1 équipé d'un adaptateur (dongle) Bluetooth VW3 A8114.

De plus, les automates compacts TWDLC•E40DRF disposent d'un port RJ45 pour la connexion réseau Ethernet qui peut être utilisé pour la communication avec un PC prenant en charge Ethernet et exécutant le logiciel de programmation TwidoSuite.

Le PC prenant en charge Ethernet peut communiquer avec le port RJ45 d'un automate Twido TWDLC•E40DRF de deux façons :

- par connexion directe via un câble inverseur UTP Ethernet RJ45 Cat5 (déconseillé) ;
- par connexion au réseau Ethernet via un câble SFTP Ethernet RJ45 Cat5 disponible dans le catalogue Schneider Electric (référence du câble : 490NTW000••).

L'automate Twido Extreme TWDLEDCK1 peut également être connecté au réseau Ethernet à l'aide d'un boîtier de connexion Modbus-Ethernet OSITRACK XGS Z33ETH.

⚠ ATTENTION

RISQUE DE DETERIORATION DU MATERIEL

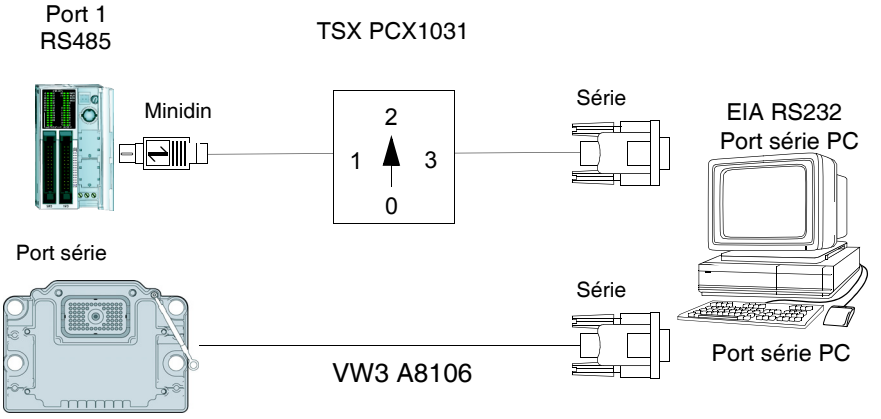
TwidoSuite risque de ne pas détecter de déconnexion lorsque vous retirez physiquement le câble de communication TSX PCX1031, TSX CRJMD25 ou Ethernet d'un automate pour le réinsérer rapidement dans un autre automate. Afin d'éviter ce genre de problème, utilisez TwidoSuite pour effectuer la déconnexion avant de retirer le câble.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Connexion par câble série

Le port EIA RS232C de votre PC peut être raccordé au port 1 de l'automate à l'aide du câble TSX PCX1031 (pour l'automate Twido Extreme TWDLEDCK1, cette connexion série est réalisée à l'aide d'un câble VW3 A8106). Le câble TSX PCX1031, qui assure la conversion des signaux entre EIA RS232 et EIA RS485, dispose d'un connecteur rotatif à 4 positions permettant de sélectionner les différents modes de fonctionnement. Les quatre positions de ce commutateur sont numérotées de 0 à 3. Pour les communications entre TwidoSuite et l'automate Twido, ce commutateur doit être positionné sur 2.

Ce raccordement est illustré dans le schéma suivant.

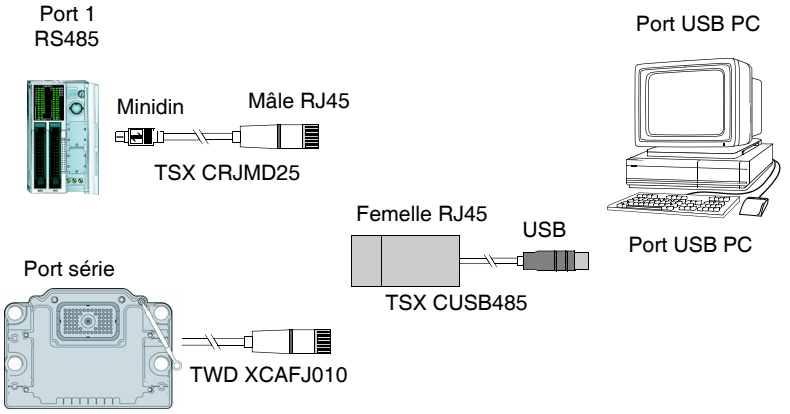


Note : Pour ce câble, le signal DPT sur la broche 5 n'est pas fixé à 0 V. Cela indique à l'automate que la connexion courante est une connexion TwidoSuite. Le signal est réglé de manière interne afin d'indiquer au microprogramme de l'automate que la connexion courante est une connexion TwidoSuite.
Pour l'automate Twido Extreme TWDLEDCK1, si le protocole ASCII est utilisé, une tension de 0 V doit être appliquée au contact de la sangle de communication (broche 22) pour permettre la communication.

Connexion par câble USB

Le port USB de votre PC peut être raccordé au port 1 de l'automate à l'aide des câbles de communication TSX CUSB485 et TSX CRJMD25 (pour l'automate Twido Extreme TWDLEDCK1, cette connexion peut être réalisée à l'aide de câbles TSX CUSB485 et TWD XCAFJ010). Le câble TSX CUSB485 convertit les signaux entre USB et EIA RS485.

Ce raccordement est illustré dans le schéma suivant.



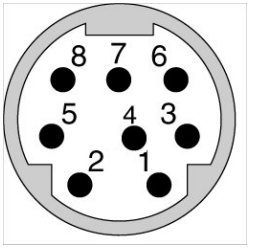
Note : Pour ce câble, le signal DPT sur la broche 5 n'est pas fixé à 0 V. Cela indique à l'automate que la connexion courante est une connexion TwidoSuite. Le signal est réglé de manière interne afin d'indiquer au microprogramme de l'automate que la connexion courante est une connexion TwidoSuite.
Pour l'automate Twido Extreme TWDLEDCK1, si le protocole ASCII est utilisé, une tension de 0 V doit être appliquée au contact de la sangle de communication (broche 22) pour permettre la communication.

Brochages des connecteurs mâle et femelle

L'illustration suivante présente le brochage d'un connecteur mini DIN mâle à 8 broches et d'un bornier :

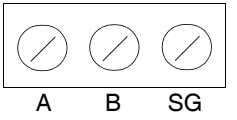
Mini DIN

TWD NAC232D, TWD NAC485D
TWD NOZ485D, TWD NOZ232D



Bornier

TWD NAC485T
TWD NOZ485T



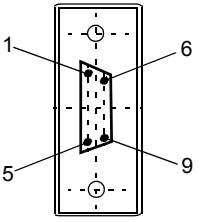
Brochages	Base RS485	Option RS485	RS232-C
1	D1 (A+)	D1 (A+)	RTS
2	D0 (B-)	D0 (B-)	DTR
3	NC	NC	TXD
4	/DE	NC	RXD
5	/DPT	NC	DSR
6	NC	NC	GND
7	0 V	0 V	GND
8	5 V	5 V	5 V

Brochages	RS485
A	D1 (A+)
B	D0 (B-)
SG	0 V

Remarque : Consommation totale maximum sur le 5 V (broche 8) : 180 mA

Remarque : Il n'y a aucune relation entre les bornier Twido RS485 A et B et les borniers D(A) et D(B) sur d'autres équipements tels que les commandes ATV, les automates XBT, Premium, etc.

L'illustration suivante présente le brochage d'un connecteur SubD femelle à 9 broches pour le TSX PCX1031.



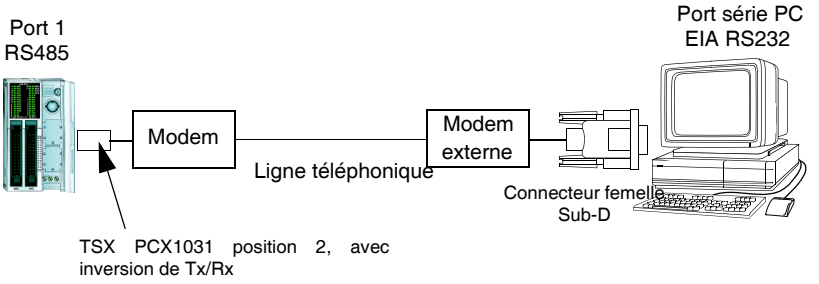
Brochages	RS232
1	DCD
2	RX
3	TX
4	DTR
5	SG
6	NC
7	RTS
8	CTS
9	NC

Connexion par ligne téléphonique

Une connexion par modem (voir *p. 83*) permet de programmer et de communiquer avec un automate par ligne téléphonique.

Le modem associé à l'automate est un modem de **réception** connecté au port 1 de l'automate. Le modem associé au PC peut être interne ou externe (alors connecté au port série COM).

Ce raccordement est illustré dans le schéma suivant.



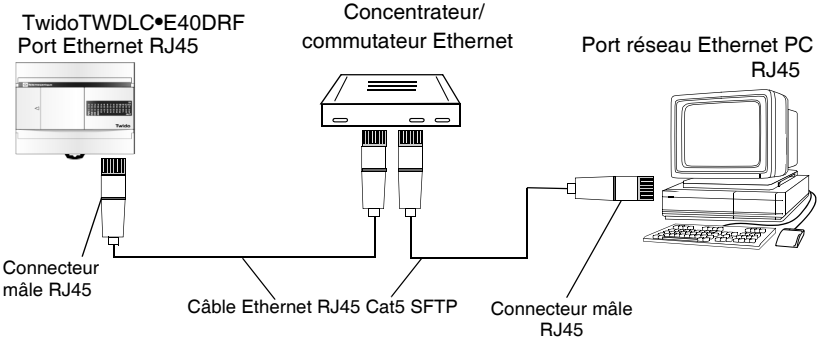
Note : Un seul modem peut être connecté au port 1 de l'automate.

Note : Attention. N'oubliez pas d'installer le logiciel fourni avec le modem, car TwidoSuite prend uniquement en compte les modems installés.

Connexion par réseau Ethernet

Note : Même si la connexion directe par câble (à l'aide d'un câble inverseur Ethernet) est prise en charge entre l'automate Twido TWDLC•E40DRF et le PC exécutant le logiciel de programmation TwidoSuite, cette méthode est déconseillée. Par conséquent, pour une connexion Ethernet, il est recommandé d'utiliser systématiquement un concentrateur/commutateur réseau Ethernet.

L'illustration suivante représente une connexion entre un PC et Twido via un concentrateur/commutateur Ethernet :



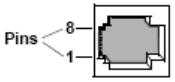
Note : Le PC exécutant l'application TwidoSuite doit prendre en charge Ethernet.

Les embases Twido TWDLC•E40DRF disposent d'un connecteur RJ45 pour la liaison au réseau Ethernet 100 BASE-TX prenant en charge l'autonégociation. Il prend en charge les vitesses de connexion réseau de 100 Mbit/s et 10 Mbit/s.

Note : Il est également possible d'utiliser un module d'interface Ethernet TwidoPort ConneXium 499TWD01100 (Vue d'ensemble du module d'interface Ethernet TwidoPort ConneXium, Twido, Guide de référence du matériel, Modules de communication) pour la connexion Ethernet. La section Configuration de TwidoPort (voir *p. 339*) explique comment déclarer et configurer ce module dans TwidoSuite.

Un exemple de connexion Ethernet pour l'automate Twido Extreme TWDLEDCK1 est proposé ci-dessous.

L'illustration suivante représente le connecteur RJ45 de l'automate Twido :



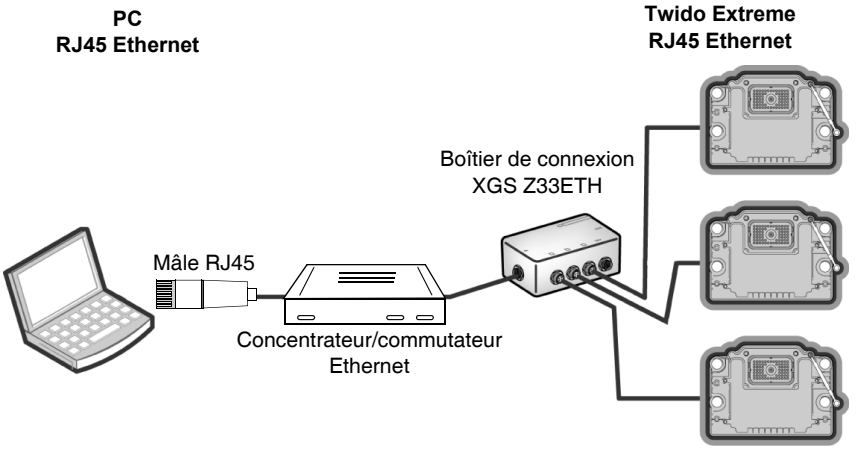
Les huit broches du connecteur RJ45 sont positionnées verticalement et numérotées par ordre croissant du bas vers le haut. Le brochage du connecteur RJ45 est décrit dans le tableau ci-dessous :

Brochage	Fonction	Polarité
8	NC	
7	NC	
6	RxD	(-)
5	NC	
4	NC	
3	RxD	(+)
2	TxD	(-)
1	TxD	(+)

Note :
- Des connecteurs et brochages identiques sont utilisés pour 10Base-T et 100Base-TX.
- Utilisez un câble Ethernet de catégorie 5 minimum pour connecter l'automate Twido à un réseau 100Base-TX.

Exemple de connexion Ethernet pour Twido Extreme

Le schéma suivant représente un exemple de connexion entre le PC exécutant le programme d'application TwidoSuite et un automate Twido Extreme TWDLEDCK1 via un boîtier de connexion XGS Z33ETH. Il est possible de connecter un maximum de 3 bases Twido Extreme sur un réseau Ethernet à l'aide de ce boîtier de connexion. Ce type de connexion est possible uniquement avec l'automate Twido Extreme TWDLEDCK1.

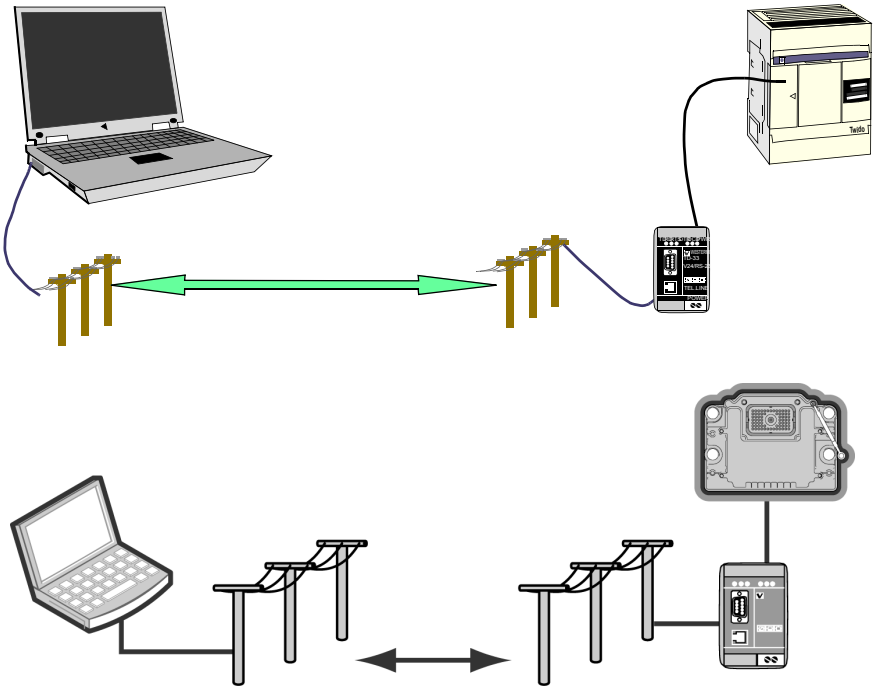


Note : Il est également possible d'utiliser un module d'interface Ethernet TwidoPort ConneXium 499TWD01100 (Vue d'ensemble du module d'interface Ethernet TwidoPort ConneXium, Twido, Guide de référence du matériel, Modules de communication) avec un câble TWD XCAFJ010 pour le raccordement à Twido Extreme. La section Configuration de TwidoPort (voir *p. 339*)explique comment déclarer et configurer ce module dans TwidoSuite.

Communication entre TwidoSuite et un modem

Généralités

Il est possible de connecter un PC exécutant TwidoSuite à un automate Twido pour transférer des applications, animer des objets et exécuter des commandes en mode opérateur. Il est également possible de connecter un automate Twido à d'autres périphériques, tels qu'un autre automate Twido, afin d'établir une communication avec le processus d'application.



Installation du modem

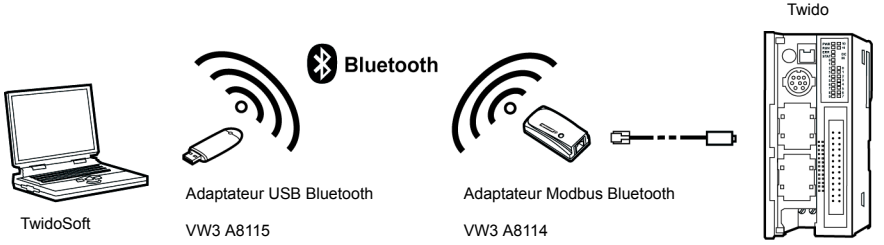
Tous les modems que l'utilisateur souhaite utiliser avec TwidoSuite doivent être installés sous l'environnement Windows à partir de votre PC.

Pour installer vos modems sous l'environnement Windows, suivez la documentation Windows.

Cette installation est indépendante de TwidoSuite.

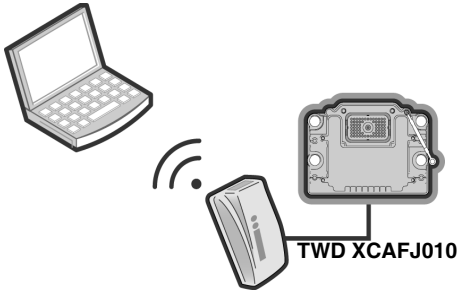
Connexion Bluetooth

Une connexion Bluetooth est également possible pour les automates Twido, via un adaptateur (verrou électronique) Bluetooth **VW3 A8114.** Les PC non équipés de la technologie Bluetooth intégrée peuvent être utilisés avec l'adaptateur Bluetooth pour PC **VW3 A8115**.



Note : Pour simplifier le téléchargement et la mise au point d'applications Twido Extreme, une connexion Bluetooth est recommandée.

Pour ce Twido Extreme, cet adaptateur Bluetooth peut être connecté au port série de l'automate à l'aide d'un câble **TWD XCAFJ010**.



Etablissement de la connexion

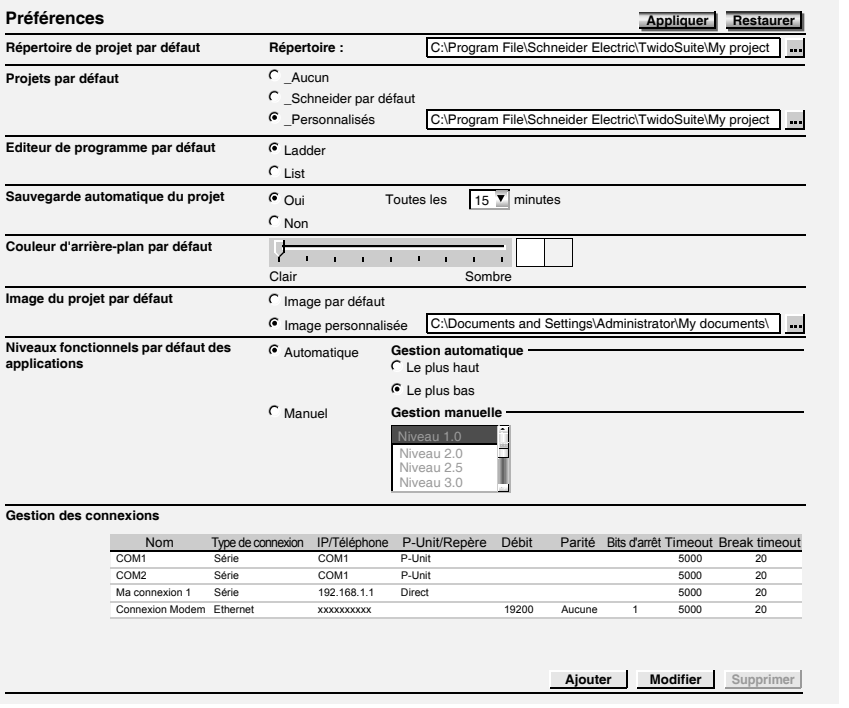
La connexion de communication par défaut entre TwidoSuite et l'automate Twido est assurée par un port de communication série, utilisant le câble TSX PCX1031 et un adaptateur croisé (reportez-vous *p. 93*). Pour l'automate TWDLEDCK1 Twido Extreme, le câble VW3 A8106 est requis pour la connexion série.

Si un modem est utilisé pour connecter le PC, alors celui-ci doit être signalé dans le logiciel TwidoSuite.

Pour sélectionner une connexion avec TwidoSuite, cliquez sur **Préférences**.

Résultat :

La boîte de dialogue suivante apparaît :



Cet écran permet de créer, modifier ou supprimer une connexion.

Pour utiliser une connexion existante, sélectionnez la connexion dans le tableau des connexions dans la tâche **Programme** → **Mise au point** → **Connecter**.

Si vous devez ajouter, modifier ou effacer une connexion, utilisez la table Gestion des connexions qui affiche la liste des connexions et leurs propriétés.

Dans ce cas, 2 ports série sont répertoriés (Com1 et Com2) et une connexion modem utilisant un modèle TOSHIBA V.90, configuré pour composer le numéro : xxxxxxxxxx (numéro de modem).

Note : Composez le numéro au format de numéro adjacent.

Vous pouvez modifier le nom de chaque connexion qui servira à la maintenance de l'application (mais le changement de COM1 ou COM2 n'est pas autorisé).

Voici le moyen de définir et sélectionner la connexion que vous voulez utiliser pour connecter votre PC à un modem.

Il ne s'agit, toutefois, qu'une partie des manipulations que vous devez effectuer pour établir la connexion globale entre le PC et l'automate Twido.

La prochaine étape concerne l'automate Twido. L'automate Twido distant doit être connecté à un modem.

Tous les modems doivent être initialisés pour établir une connexion. L'automate Twido, équipé au minimum du microprogramme version V2.0, est capable d'envoyer à la mise sous tension une chaîne adaptée au modem, si le modem est configuré dans l'application.

Configuration du modem

Pour configurer un modem dans un automate Twido, procédez comme suit :

Pour ajouter un modem à une application ouverte, suivez la procédure décrite dans la rubrique (Positionnement d'un modem, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).

Après avoir configuré le modem sur le port 1, vous devez définir les propriétés. Dans l'étape Description, cliquez deux fois sur l'icône Modem .

Résultat : la boîte de dialogue Caractéristiques du modem (Configuration du modem, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) s'affiche. La boîte de dialogue Propriétés du Modem permet de sélectionner un modem existant, d'en créer un nouveau ou de modifier la configuration d'un modem.

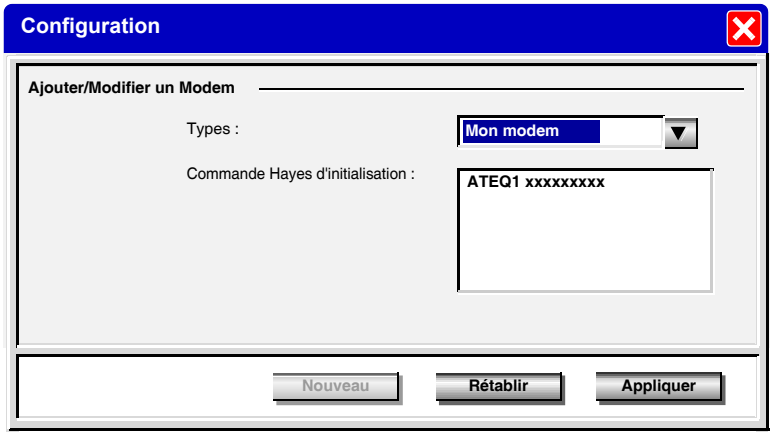
Illustration de la boîte de dialogue Caractéristiques du modem :

La configuration sélectionnée correspond à celle lue dans l'automate : La commande Hayes d'initialisation, alors lue, est affichée dans le format Hayes standard.

Note : La gestion du modem par l'automate Twido est effectuée sur le port 1. Cela signifie que vous pouvez connecter un modem sur le port de communication 2, mais dans ce cas, tous les modes opératoires et la séquence d'initialisation du modem doivent être effectués manuellement et non de la même manière qu'avec le port de communication 1 (l'automate TWDLEDCK1 Twido Extreme possède un seul port série).

Vous pouvez sélectionner un modem défini précédemment ou en créer un nouveau en cliquant sur "Nouveau".

Illustration de la boîte de dialogue Ajouter/Modifier un modem :



Attribuez ensuite un nom au nouveau profil et remplissez la commande Hayes d'initialisation comme décrit dans la documentation du modem.

Sur cette illustration, "xxxxxx" représente la séquence d'initialisation que vous devez saisir afin de préparer le modem à la communication, c'est-à-dire le débit, la parité, le bip d'arrêt et le mode de réception.

Pour compléter cette séquence vous devez vous référer à la documentation de votre modem.

La longueur maximum de la chaîne est : 127 caractères.

Lorsque votre application est terminée ou, au minimum, lorsque le port 1 de communication est totalement configuré, transférez l'application en utilisant une "connexion point à point".

L'automate Twido est alors prêt à être connecté à un PC exécutant TwidoSuite par l'intermédiaire de modems.

Séquence de connexion

Après avoir préparé TwidoSuite et l'automate Twido, établissez la connexion comme suit :

Etape	Action
1	Mettez sous tension l'automate Twido et le modem.
2	Démarrez votre PC et lancez TwidoSuite.
3	Utilisez la tâche **Préférences** et sélectionnez une connexion modem dans la table "Gestion des connexions" (par exemple, "Mon modem" ou le nom que vous avez donné à votre connexion modem : reportez-vous à la section "création d'une connexion").
4	Connectez TwidoSuite

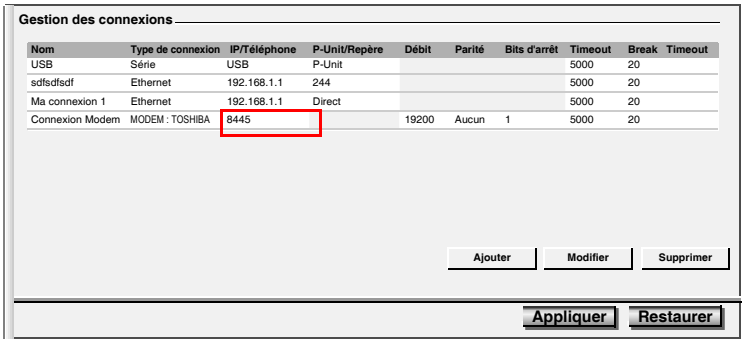
Note : Si vous voulez toujours utiliser votre connexion modem, sélectionnez "mon modem" dans les Préférences du menu Fichier (ou le nom que vous lui avez donné). Ainsi, TwidoSuite mémorise cette préférence.

Modes de marche

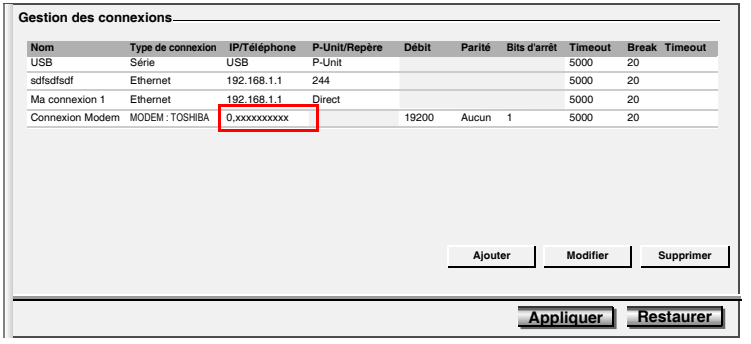
L'automate Twido envoie la chaîne d'initialisation au modem connecté sous tension. Lorsqu'un modem est configuré dans l'application Twido, l'automate envoie d'abord une commande "AT&F" afin de savoir si le modem est connecté. Si l'automate reçoit une réponse, alors la chaîne d'initialisation est envoyée au modem.

Appel Interne, Externe et International

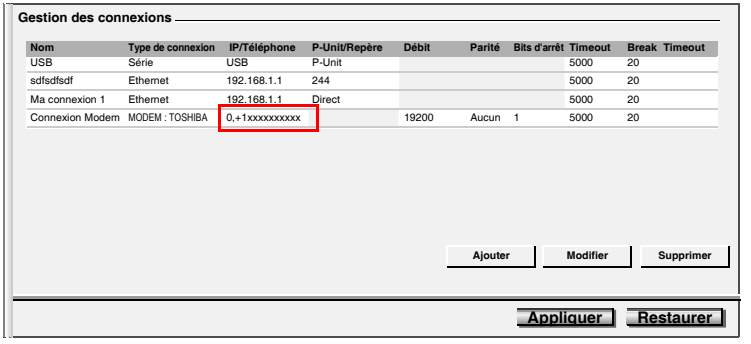
Si vous communiquez avec un automate Twido dans l'enceinte de votre entreprise, vous pouvez seulement utiliser l'extension de ligne que vous devez composer, comme : 8445


Gestion des connexions
Nom Type de connexion IP/Téléphone P-Unit/Repère Débit Parité Bits d'arrêt Timeout Break Timeout
USB Série USB P-Unit 5000 20
sdfsdfsdf Ethernet 192.168.1.1 244 5000 20
Ma connexion 1 Ethernet 192.168.1.1 Direct 5000 20
Connexion Modem MODEM : TOSHIBA 8445 19200 Aucun 1 5000 20
Ajouter
Modifier
Supprimer
Appliquer
Restaurer

Si vous utilisez un standard interne pour composer les numéros de téléphone en dehors de votre entreprise et que vous devez composer un "0" ou un "9" avant le numéro de téléphone utilisez la syntaxe suivante : 0,xxxxxxxxxx ou 9,xxxxxxxxxx


Gestion des connexions
Nom Type de connexion IP/Téléphone P-Unit/Repère Débit Parité Bits d'arrêt Timeout Break Timeout
USB Série USB P-Unit 5000 20
sdfsdfsdf Ethernet 192.168.1.1 244 5000 20
Ma connexion 1 Ethernet 192.168.1.1 Direct 5000 20
Connexion Modem MODEM : TOSHIBA 0,xxxxxxxxxx 19200 Aucun 1 5000 20
Ajouter
Modifier
Supprimer
Appliquer
Restaurer

Pour les appels internationaux la syntaxe est : +1xxxxxxxxxx, par exemple. Et si vous utilisez un standard : 0,+1xxxxxxxxxx


Gestion des connexions
Nom Type de connexion IP/Téléphone P-Unit/Repère Débit Parité Bits d'arrêt Timeout Break Timeout
USB Série USB P-Unit 5000 20
sdfsdfsdf Ethernet 192.168.1.1 244 5000 20
Ma connexion 1 Ethernet 192.168.1.1 Direct 5000 20
Connexion Modem MODEM : TOSHIBA 0,+1xxxxxxxxxx 19200 Aucun 1 5000 20
Ajouter
Modifier
Supprimer
Appliquer
Restaurer

Questions fréquemment posées

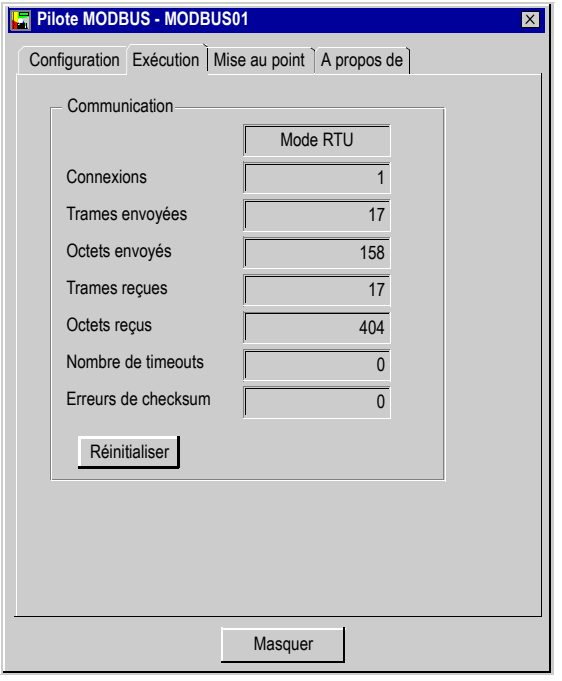
Lorsque votre communication est établie depuis quelques minutes, des erreurs de communication peuvent survenir. Dans ce cas, vous devez régler les paramètres de communication.

TwidoSuite utilise un pilote Modbus pour communiquer via des ports série ou des modems internes. Dès que la communication est établie, Pilote Modbus apparaît dans la barre d'outils. Cliquez deux fois sur l'icône Pilote Modbus pour ouvrir la fenêtre. Vous avez désormais accès aux paramètres Pilote Modbus et l'onglet "Exécution" affiche des informations sur les trames échangées avec l'automate à distance.

Si la valeur Nombre de timeouts augmente ou est différente de 0, modifiez-la à l'aide de la table Gestion des connexions, accessible sous TwidoSuite en utilisant la tâche

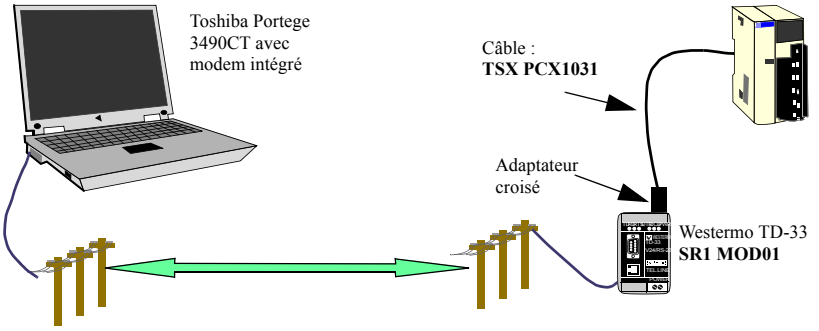
Préférences. Cliquez sur le champ Timeout, puis sur le bouton de modification et entrez une valeur plus élevée. La valeur par défaut est "5 000" (en millième de secondes).

Essayez ensuite de vous reconnecter. Réglez la valeur jusqu'à ce que votre connexion soit stable.



Exemples

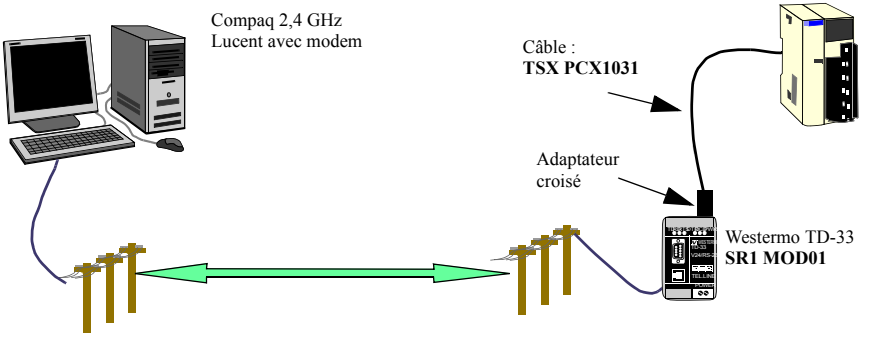
- **Exemple 1** : TwidoSuite connecté à un TWD LMDA 20DRT (Windows 98 SE).
  - PC : Toshiba Portege 3490CT sous Windows 98,
  - Modem (interne au PC) : Toshiba internal V.90,
  - Automate Twido : TWD LMDA 20DRT version 2.0,
  - Modem (connecté à l'automate Twido) : Type Westermo TD-33 / V.90 référence SR1 MOD01 disponible sur le nouveau catalogue Twido (Septembre 03) (reportez-vous à l'*Annexe 2, p. 94*), (**clients nord-américains uniquement :** le type de modem disponible dans votre région est le TD-33/V.90 US),
  - Câble : TSX PCX 1031 connecté au port de communication Twido 1 et un adaptateur : 9 broches mâle / 9 broches mâle afin de croiser Rx et Tx durant la connexion entre le modem Westermo et l'automate Twido (reportez-vous à la section *Annexe 1, p. 93*). Vous pouvez également utiliser le câble TSX PCX 1130 (conversion RS485/232 et croisement Rx/Tx).



Le premier test consiste à utiliser 2 lignes de téléphone analogiques, internes à l'entreprise, n'utilisant pas le numéro de téléphone complet, mais juste l'extension (c'est pourquoi il n'y a que 4 chiffres pour le numéro de téléphone de modem Toshiba V.90 interne).

Pour ce test, les paramètres de connexion (TwidoSuite, Préférences, puis Gestion des connexions) étaient définis sur leur valeur par défaut, soit Timeout = 5 000 et Break timeout = 20.

- **Exemple 2** : TwidoSuite connecté à TWD LMDA 20DRT (Windows XP Pro)
  - PC : Compaq Pentium 4, 2,4 GHz,
  - Modem : Lucent Win modem, bus PCI,
  - Automate Twido : TWD LMDA 20DRT version 2.0,
  - Modem (connecté à l'automate Twido) : Type WESTERMO TD-33 / V.90 référence SR1 MOD01 disponible dans le nouveau catalogue Twido (Septembre 03) (reportez-vous à l'*Annexe 2, p. 94*), (**clients nord-américains uniquement :** le type de modem disponible dans votre région est le TD-33/V.90 US),
  - Câble : TSX PCX 1031 connecté au port 1 de communication Twido et un adaptateur : 9 broches mâle / 9 broches mâle afin de croiser Rx et Tx durant la connexion entre le modem Westermo et l'automate Twido (reportez-vous à la section *Annexe 1, p. 93*). Vous pouvez également utiliser le câble TSX PCX 1130 (conversion RS485/232 et croisement Rx/Tx).

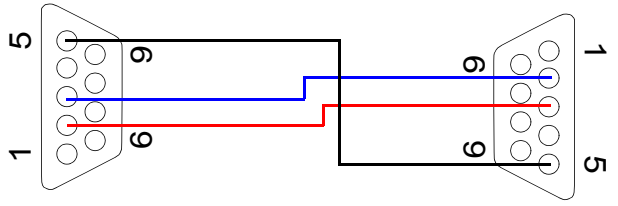


Le test consiste à utiliser deux lignes de téléphone analogiques, internes à l'entreprise, n'utilisant pas le numéro de téléphone complet, mais juste l'extension (C'est pourquoi il y n'a que 4 chiffres pour le numéro de téléphone du modem interne Toshiba V.90).

Pour ce test, les paramètres de connexion (TwidoSuite, Préférences, puis Gestion des connexions) étaient définis sur leur valeur par défaut, soit Timeout = 5 000 et Break timeout = 20.

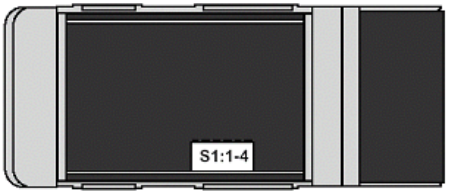
Annexe 1

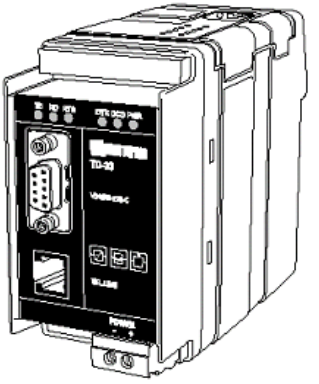
Adaptateur croisé pour le câble TSX PCX 1031 et modem Westermo TD-33 :



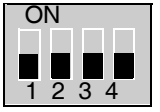
Annexe 2

Modem Westermo TD-33, référence Schneider **SR1 MOD01**(1). Ce modem gère quatre interrupteurs DIP, qui doivent tous être définis sur **OFF** :





Réglages usine



Utiliser la configuration stockée (vitesse et format, etc.)
Désactiver DTR Hotcall, Auto Baud

Note :

1. Certains produits peuvent ne pas être compatibles et/ou disponibles dans toutes les régions. Pour plus d'informations, contactez votre représentant Schneider local.

Annexe 3

Modem Wavecom WMOD2B, référence Schneider **SR1 MOD02**(1) double bande (900/1 800 Hz) :

Note :
1. Certains produits peuvent ne pas être compatibles et/ou disponibles dans toutes les régions. Pour plus d'informations, contactez votre représentant Schneider local.

Annexe 4

Références des produits utilisés dans ce document :

- Produit Twido : TWD LMDA 20DRT
- Logiciel TwidoSuite : TWD SPU 1002 V10M
- Câble TSX PCX1031
- Câble TSX PCX1130
- Modem RTU : Westermo TD-33 / V90 **SR1 MOD01**(1),
- Modem GSM : Wavecom WMOD2B **SR1 MOD02**(1).

Note :
1. Certains produits peuvent ne pas être compatibles et/ou disponibles dans toutes les régions. Pour plus d'informations, contactez votre représentant Schneider local.

Communications par liaison distante

Introduction

La liaison distante est un bus maître/esclave à haut débit conçu pour assurer l'échange d'une petite quantité de données entre l'automate maître et un maximum de sept automates (esclaves) distants. Les données de l'application ou les données d'E/S sont transférées en fonction de la configuration des automates distants. Il est possible d'associer différents types d'automates, tels que des automates d'E/S distantes et des automates d'extension.

Note : Les communications par liaison distante ne sont pas prises en charge par l'automate Twido Extreme TWDLEDCK1.

Note : L'automate maître contient les informations relatives au repère d'une E/S distante, mais il ne sait pas à quel automate précis correspond ce repère. Par conséquent, l'automate maître ne peut pas affirmer que toutes les entrées et sorties distantes utilisées dans l'application utilisateur existent réellement. Vous devez donc vous assurer qu'elles existent réellement.

Note : Le bus d'E/S distantes et le protocole utilisé sont propriétaires et aucun équipement tiers n'est autorisé sur le réseau.

⚠ ATTENTION

FONCTIONNEMENT INATTENDU DU MATERIEL

- Assurez-vous qu'il existe un seul automate maître sur une liaison distante et que chaque esclave dispose d'un repère unique. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.
- Assurez-vous que tous les esclaves disposent d'un repère unique. Deux esclaves ne doivent pas avoir le même repère. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Note : La liaison distante nécessite une connexion EIA RS485 et peut être exécutée sur un seul port de communication à la fois.

Configuration matérielle

Une liaison distante doit utiliser un port EIA RS485 à 3 fils minimum. Il est possible de la configurer afin d'utiliser le premier port ou un deuxième port optionnel existant.

Note : Seul un seul port de communication à la fois peut être configuré en tant que liaison distante.

Le tableau suivant répertorie les équipements qui peuvent être utilisés :

Périphérique	Port	Caractéristiques
TWDLC•A10/16/24DRF, TWDLC••40DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Base automate équipée d'un port EIA RS485 à 3 fils avec un connecteur mini DIN.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion de l'afficheur.
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN ou d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Communication.

Note : La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate.

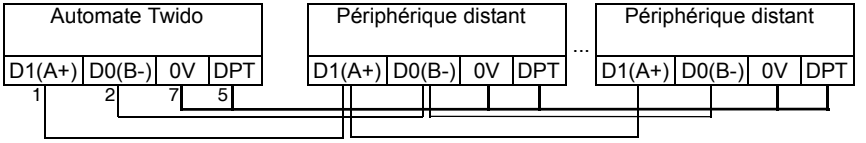
Connexion de câbles à chaque équipement

Note : Le signal DPT sur la broche 5 doit être relié au 0 V sur la broche 7 afin de signaler l'utilisation de communications par liaison distante. Lorsque ce signal n'est pas relié à la terre, l'automate Twido maître ou esclave est défini par défaut sur un mode via lequel des tentatives d'établissement de communication avec TwidoSuite sont effectuées.

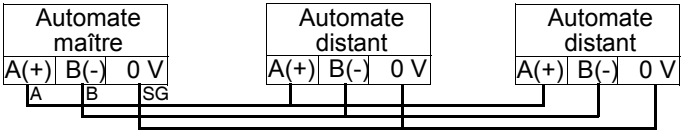
Note : La connexion DPT à 0 V n'est nécessaire qu'en cas de connexion à une base automate sur le port 1.

Les connexions de câbles effectuées à chaque équipement distant sont représentées ci-dessous.

Connexion mini DIN



Connexion bornier



Configuration logicielle

Un seul automate maître doit être défini sur la liaison distante. En outre, chaque automate distant doit conserver un repère esclave unique. L'utilisation de repères identiques par plusieurs maîtres ou esclaves risque d'altérer des transmissions ou de créer des ambiguïtés.

⚠ ATTENTION

FONCTIONNEMENT INATTENDU DE L'ÉQUIPEMENT

Assurez-vous qu'il existe un seul automate maître sur une liaison distante et que chaque esclave dispose d'un repère unique. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Configuration de l'automate maître

Configurez l'automate maître à l'aide de TwidoSuite pour gérer un réseau de liaison distante constitué au maximum de sept automates distants. Ces sept automates distants peuvent être configurés en tant qu'E/S distantes ou automates d'extension. Le repère du maître configuré à l'aide de TwidoSuite correspond au repère 0.

Pour configurer un automate en tant que maître, vous devez configurer le port 1 ou le port 2 en liaison distante et choisir le repère 0 (maître) à l'aide de TwidoSuite.

Puis, dans la fenêtre "Ajouter un automate distant", définissez les automates esclaves en E/S distantes ou en automates d'extension, ainsi que leurs repères.

Configuration de l'automate distant

La configuration d'un automate distant s'effectue à l'aide de TwidoSuite en configurant le port 1 ou 2 en liaison distante et ou en affectant au port un repère compris entre 1 et 7.

Le tableau suivant résume les différences et les contraintes de chacun de ces types de configuration d'automate distant :

Type	Programme d'application	Accès aux données
E/S distantes	Non Pas même une simple instruction "END". Le mode RUN dépend de celui du maître.	%I et %Q Seule l'E/S locale de l'automate distant est accessible (et non son extension d'E/S).
Automate d'extension	Oui Le mode RUN est indépendant de celui du maître.	%INW et %QNW Il est possible de transmettre un maximum de quatre mots d'entrée et quatre mots de sortie vers et depuis chaque extension.

Synchronisation de scrutation de l'automate distant

Le cycle de mise à jour de la liaison distante n'est pas synchronisé avec la scrutation de l'automate maître. Les communications avec les automates distants sont déclenchées par interruption et se produisent en tant que tâches d'arrière-plan, en parallèle avec l'exécution de la scrutation de l'automate maître. A la fin du cycle de scrutation, les valeurs les plus récentes sont lues dans les données d'application à utiliser pour la prochaine exécution de programme. Ce traitement est le même pour les automates d'E/S distantes et d'extension.

Tous les automates peuvent vérifier l'activité de la liaison générale à l'aide du bit système %S111. Mais pour accomplir la synchronisation, un automate maître ou d'extension doit utiliser le bit système %S110. Ce bit est mis à 1 une fois qu'un cycle de mise à jour complet s'est déroulé. Le programme d'application est responsable de sa remise à 0.

Le maître peut activer ou désactiver la liaison distante à l'aide du bit système %S112. Les automates peuvent contrôler la configuration et le bon fonctionnement de la liaison distante à l'aide du bit %S113. Le signal DPT sur le port 1 (utilisé pour déterminer si TwidoSuite est connecté) est détecté et signalé sur %S100.

Le tableau suivant résume toutes ces informations :

Bit système	Etat	Indication
%S100	0	Maître/esclave : DPT inactif (câble TwidoSuite NON connecté)
	1	Maître/esclave : DPT actif (câble TwidoSuite connecté)
%S110	0	Maître/esclave : mis à 0 par l'application
	1	Maître : tous les échanges de liaison distante effectués (E/S distantes uniquement) Esclave : échange avec le maître effectué
%S111	0	Maître : échange de liaison distante unique effectué Esclave : échange de liaison distante unique détecté
	1	Maître : échange de liaison distante unique en cours Esclave : échange de liaison distante unique détecté
%S112	0	Maître : communication par liaison distante activée
	1	Maître : communication par liaison distante désactivée
%S113	0	Maître/esclave : configuration/fonctionnement de la liaison distante OK
	1	Maître : erreur de configuration/fonctionnement de la liaison distante Esclave : erreur de fonctionnement de la liaison distante

Redémarrage de l'automate maître

Lorsqu'un automate maître redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, le maître continue de communiquer avec les esclaves.

Redémarrage de l'automate esclave

Lorsqu'un automate esclave redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'esclave continue de communiquer avec le maître. Si le maître indique un état Stop :
  - Les E/S distantes appliquent un état Stop.
  - L'automate d'extension continue dans son état actuel.

Arrêt de l'automate maître

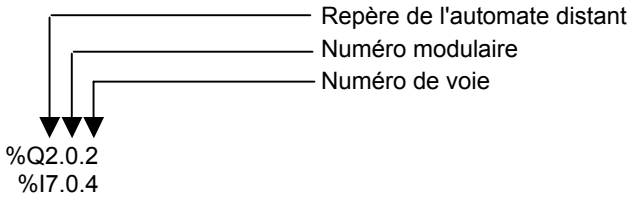
Lorsque l'automate maître passe en mode Stop, tous les équipements esclaves continuent de communiquer avec le maître. Lorsque le maître indique qu'un arrêt est requis, un automate d'E/S distantes s'arrête, mais les automates d'extension continuent dans leur état courant Run ou Stop.

Accès aux données E/S distantes

L'automate distant configuré en tant qu'E/S distantes ne possède, ni n'exécute son propre programme d'application. Les entrées et sorties TOR de base de l'automate distant sont une simple extension de celles de l'automate maître. L'application doit uniquement utiliser le mécanisme de repérage complet à trois chiffres fourni.

Note : Le numéro de module est toujours zéro pour les E/S distantes.

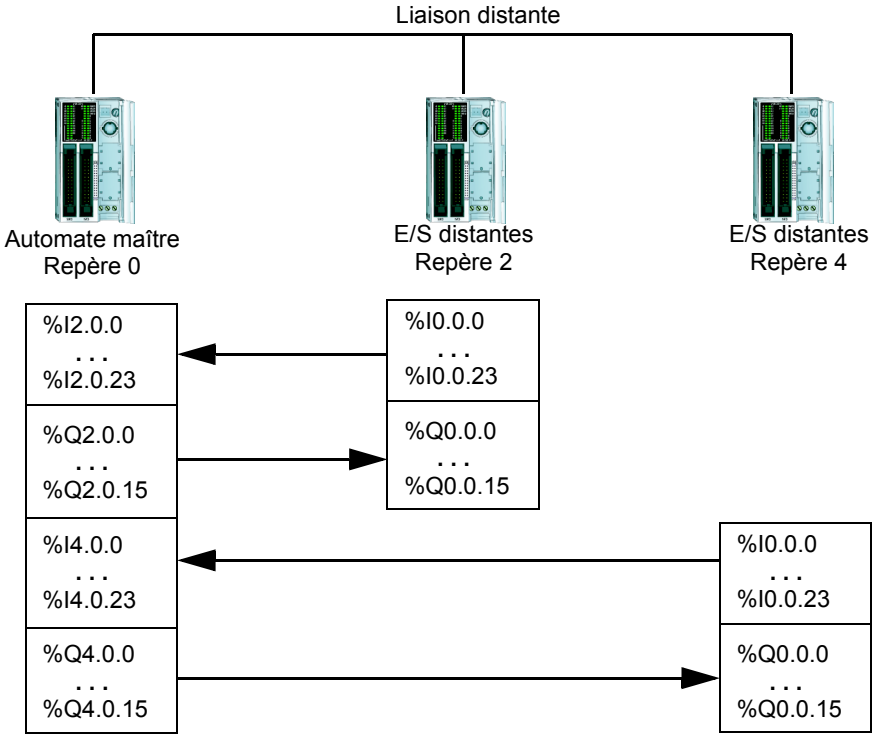
Illustration



Pour communiquer avec les E/S distantes, l'automate maître utilise la notation d'entrée et sortie standard %I et %Q. Pour accéder au troisième bit de sortie de l'E/S distante configurée au repère 2, on utilise l'instruction %Q2.0.2. De même, pour lire le cinquième bit d'entrée de l'E/S distante configurée au repère 7, on utilise l'instruction %I7.0.4.

Note : L'accès du maître est restreint aux E/S TOR appartenant aux E/S locales de l'automate distant. Aucune E/S analogique ou d'expansion ne peut être transférée, hormis en cas d'utilisation de communications d'extension.

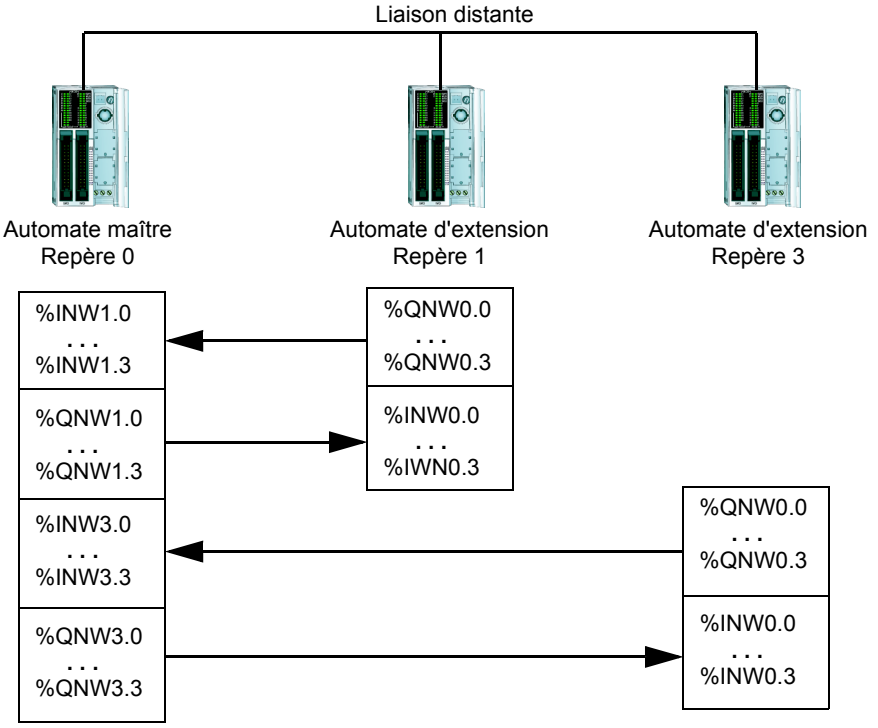
Illustration



Accès aux données de l'automate d'extension

Pour communiquer avec des automates d'extension, le maître utilise les mots réseau %INW et %QNW afin d'échanger des données. Chaque extension du réseau est accessible par son repère distant "j" à l'aide de mots %INWj.k et %QNWj.k. Chaque automate d'extension du réseau utilise %INW0.0 à %INW0.3 et %QNW0.0 à %QNW0.3 pour accéder aux données situées sur le maître. Les mots réseau sont automatiquement mis à jour lorsque les automates sont en mode Run ou Stop.

L'exemple suivant illustre l'échange d'un maître avec deux automates d'extension configurés.



Il n'existe aucune remise de messages de poste à poste au sein de la liaison distante. Il est possible d'utiliser le programme d'application du maître pour gérer les mots réseaux, afin de transférer des informations entre des automates distants qui utilisent alors le maître en tant que passerelle.

Informations d'état

Outre les bits système décrits précédemment, le maître conserve l'état de présence et de configuration des automates distants. Cette action s'effectue dans les mots systèmes %SW111 et %SW113. L'automate maître ou l'automate distant peut obtenir la valeur de la dernière erreur survenue pendant la communication sur la liaison distante dans le mot système %SW112.

Mots système	**Utilisation**	
%SW111	Etat de la liaison distante : deux bits pour chaque automate distant (maître uniquement)	
	x0-6	0 - automate distant 1-7 absent
		1 - automate distant 1-7 présent
	x8-14	0 - E/S distante détectée sur l'automate distant 1-7
		1 - automate d'extension détecté sur l'automate distant 1-7
%SW112	Code d'erreur de configuration ou de fonctionnement de la liaison distante	
		00 : opérations réussies
		01 - expiration du timeout (esclave)
		02 - erreur de checksum détectée (esclave)
		03 - incohérence de configuration (esclave)
		04 - (pour le port 1 uniquement) port indisponible, P-Unit connecté ou mode P-Unit
%SW113	Configuration de la liaison distante : deux bits pour chaque automate distant (maître uniquement)	
	x0-6	0 - automate distant 1-7 non configuré
		1 - automate distant 1-7 configuré
	x8-14	0 - E/S distante configurée en tant qu'automate distant 1-7
		1 - automate d'extension configuré en tant qu'automate distant 1-7

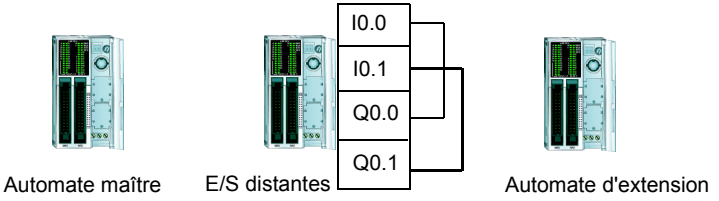
Exemple de liaison distante

Pour configurer une liaison distante, procédez comme suit :

1. Configurez le matériel.
2. Câblez les automates.
3. Connectez le câble de communication entre le PC et les automates.
4. Configurez le logiciel.
5. Ecrivez une application.

Les illustrations suivantes représentent une utilisation de la liaison distante avec les E/S distantes et un automate d'extension.

Etape 1 : Configuration du matériel :

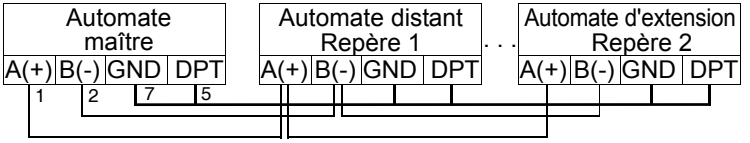


La configuration matérielle comprend trois bases automates de tout type. Le port 1 est utilisé selon deux modes de communication. L'un des modes permet de configurer et de transférer le programme d'application à l'aide de TwidoSuite. Le second mode est destiné au réseau de liaison distante. Si un port 2 optionnel est disponible sur l'un des automates, il est possible de l'utiliser, mais un automate ne gère qu'une seule liaison distante.

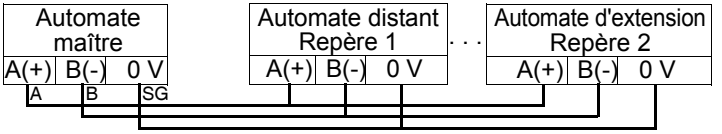
Note : Dans cet exemple, les deux premières entrées sur les E/S distantes sont câblées sur les deux premières sorties.

Etape 2 : Câblage des automates :

Connexion mini DIN

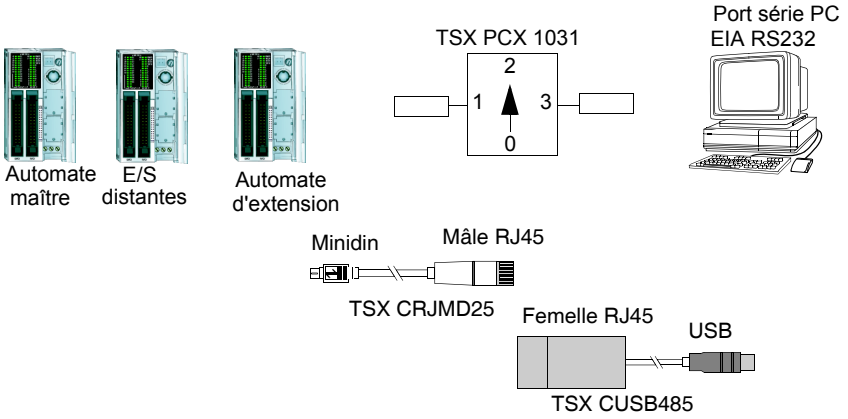


Connexion bornier



Connectez les câbles des signaux A(+) et B(-) ensemble. Sur chaque automate, le signal DPT est relié à la terre. Bien que la mise à la terre du signal ne soit pas obligatoire pour une utilisation avec une liaison distante sur le port 2 (cartouche ou module de communication optionnel), il s'agit d'une bonne habitude à prendre.

Etape 3 : Connexion du câble de communication entre le PC et les automates :



Les câbles de programmation multifonctions TSX PCX1031 ou TSX CUSB485 et TSX CRJMD25 sont utilisés pour communiquer avec chacune des trois bases automates. Assurez-vous que le commutateur du câble est en position 2. Afin de programmer chaque automate, il est nécessaire d'établir une communication point à point avec chaque automate. Pour établir cette communication, connectez-vous au port 1 du premier automate, transférez la configuration et les données de l'application, puis mettez l'automate en Run. Répétez cette procédure pour chaque automate.

Note : Il est nécessaire de déplacer le câble après chaque configuration d'automate et transfert d'application.

Etape 4 : Configuration du logiciel :

Chacun des trois automates utilise TwidoSuite pour créer une configuration, et le cas échéant, le programme d'application.

Pour l'automate maître, éditez le paramétrage des communications de l'automate afin de régler le protocole sur "Liaison distante" et le repère sur "0 (Maître)".

Paramétrage des comm. de l'automate
Type : Liaison distante
Repère : 0 (maître)

Configurez l'automate distant sur le maître en ajoutant une "E/S distante" au repère "1" et un "Automate d'extension" au repère "2".

Ajouter automates distants

Utilisation automate : E/S distantes
Repère distant : 1

Utilisation automate : Automate d'extension
Repère distant : 2

Pour l'automate configuré en tant qu'E/S distantes, vérifiez que la configuration de la communication de l'automate est réglée sur "Liaison distante" et le repère sur "1".

Paramétrage des comm. de l'automate
Type : Liaison distante
Repère : 1

Pour l'automate configuré en tant qu'extension, vérifiez que la configuration de la communication de l'automate est réglée sur "Liaison distante" et sur le repère "2".

Paramétrage des comm. de l'automate
Type : Liaison distante
Repère : 2

Etape 5 : Ecriture des applications :

Pour l'automate maître, écrivez le code du programme d'application suivant :

```
LD 1

[%MW0 := %MW0 +1]
[%QNW2.0 := %MW0]
[%MW1 := %INW2.0]

LD %I0.0
ST %Q1.00.0
LD %I1.0.0
ST %Q0.0

LD %I0.1
ST %Q1.0.1
LD %I1.0.1
ST %Q0.1
```

Pour l'automate configuré en tant qu'E/S distantes, n'écrivez pas de programme d'application.

Pour l'automate configuré en tant qu'extension, écrivez l'application suivante :

```
LD 1
[%QNW0.0 := %INW0.0]
```

Dans cet exemple, l'application maître incrémente un mot mémoire interne et le communique à l'automate d'extension à l'aide d'un seul mot réseau. L'automate d'extension prend le mot reçu du maître et le renvoie. Dans le maître, un mot mémoire différent reçoit et stocke cette transmission.

Pour communiquer avec l'automate d'E/S distantes, le maître envoie ses entrées locales aux sorties des E/S distantes. A l'aide de la connexion E/S externe des E/S distantes, les signaux sont renvoyés et récupérés par le maître.

Communications ASCII

Introduction

Le protocole ASCII offre aux automates Twido un protocole de mode caractère semi-duplex simple permettant d'émettre et/ou de recevoir des données à l'aide d'un seul équipement. Ce protocole est pris en charge à l'aide de l'instruction EXCHx et géré à l'aide du bloc fonction %MSGx.

Les trois types de communications suivants sont possibles à l'aide du protocole ASCII :

- Emission seule
- Emission/réception
- Réception seule

La taille maximale des trames émises et/ou reçues à l'aide de l'instruction EXCHx s'élève à 256 octets.

Configuration matérielle

Il est possible d'établir une liaison ASCII (voir les bits systèmes %S103 et %S104 (voir *p. 674*)) sur le port EIA RS232 ou EIA RS485 et de l'exécuter simultanément sur deux ports de communication au maximum.

Le tableau suivant répertorie les équipements qui peuvent être utilisés :

Equipement	Port	Caractéristiques
TWDLC•A10/16/24DRF, TWDLC••40DRF, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Base automate équipée d'un port EIA RS485 à 3 fils avec un connecteur mini DIN.
TWDLEDCK1	1	Base automate équipée d'un port type EIA RS485 non isolé, longueur maximale limitée à 200 m. **Remarque** : Les options de configuration suivantes ne sont **pas possibles**. • 7 bits, pas de parité, 1 bit d'arrêt • 8 bits, parité paire, 2 bits d'arrêt • 8 bits, parité impaire, 2 bits d'arrêt
TWDNOZ232D	2	Module de communication équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.

Equipement	Port	Caractéristiques
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNAC232D	2	Adaptateur de communication équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN, d'un port EIA RS485 à 3 fils avec un connecteur mini DIN et d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Communication.

Note : La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate. (L'automate Twido Extreme TWDLEDCK1 ne possède qu'un port série.)

Câblage nominal Les connexions de câble nominal sont représentées ci-dessous pour les types EIA RS232 et EIA RS485.

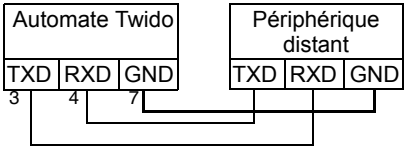
Note : Si le port 1 est utilisé sur l'automate Twido, le signal DPT sur la broche 5 doit être relié au 0 V sur la broche 7. Ce signal permet d'indiquer à l'automate Twido que les communications via le port 1 relèvent du protocole ASCII et non du protocole utilisé pour communiquer avec le logiciel TwidoSuite.

Note : Pour l'automate Twido Extreme TWDLEDCK1, si le protocole ASCII est utilisé, une tension de 0 V doit être appliquée au contact de la sangle de communication (broche 22) pour permettre la communication. Cela indique à l'automate Twido que la communication via le port 1 n'est pas le protocole utilisé pour communiquer avec le logiciel TwidoSuite.

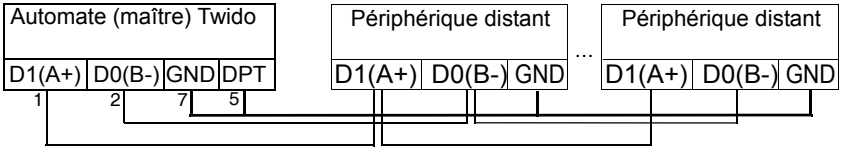
Les connexions de câbles effectuées à chaque équipement sont représentées ci-dessous.

Connexion mini DIN

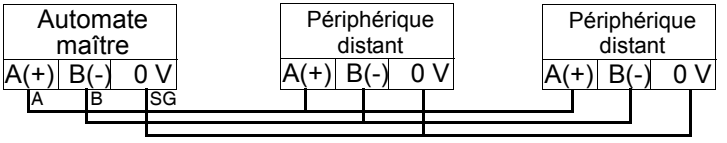
Câble RS232 EIA



Câble RS485 EIA



Connexion bornier



Configuration logicielle

Pour configurer l'automate afin d'utiliser une liaison série pour envoyer et recevoir des caractères à l'aide du protocole ASCII, procédez comme suit :

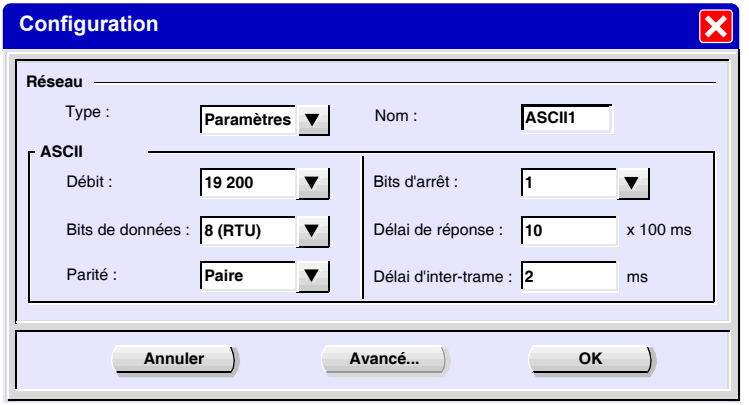
Etape	Description
1	Configurez le port série pour le protocole ASCII à l'aide de TwidoSuite[1].
2	Créez dans votre application une table d'émission/réception qui sera utilisée par l'instruction EXCHx.

[1]Les options de configuration suivantes ne sont pas possibles pour l'automate Twido Extreme TWDLEDCK1 :

- **7 bits, pas de parité, 1 bit d'arrêt**
- **8 bits, parité paire, 2 bits d'arrêt**
- **8 bits, parité impaire, 2 bits d'arrêt**

Configuration du port

Un automate Twido peut utiliser son port 1 principal ou un port 2 configuré en option pour utiliser le protocole ASCII (l'automate Twido Extreme TWDLEDCK1 ne possède qu'un port série). Pour configurer un port série pour le protocole ASCII :

Etape	Action
1	Définissez tous les modules ou adaptateurs de communication supplémentaires configurés dans la base.
2	Déclarez le réseau ASCII dans l'étape Description de TwidoSuite (reportez-vous aux sections (Comment créer un réseau (exemple Modbus), Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et (Comment créer un réseau (exemple Modbus), Twido, Logiciel de programmation TwidoSuite, Aide en ligne) pour le protocole ASCII).
3	Sélectionnez le **port 1** (ou le **port 2** s'il est installé) pour configurer dans la fenêtre Description (reportez-vous à la section (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)).
4	Pour configurer l'élément ASCII, utilisez l'une des deux méthodes suivantes : • cliquez sur l'icône Configurer dans la barre d'outils, puis sélectionnez l'élément ASCII dans le graphique descriptif ; • cliquez deux fois sur l'élément ASCII dans le graphique descriptif.
5	Utilisez l'une des deux méthodes suivantes pour ouvrir la boîte de dialogue Caractéristique (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) des paramètres matériels de liaison ASCII : • cliquez sur l'icône Configurer dans la barre d'outils, puis sélectionnez la liaison ASCII dans le graphique descriptif ; • cliquez deux fois sur la liaison ASCII dans le graphique descriptif.
6	Configurez la boîte de dialogue Caractéristique qui s'affiche, en procédant comme suit : 
7	Définissez les paramètres de communication. Les options de configuration suivantes ne sont **pas possibles** pour l'automate Twido Extreme TWDLEDCK1 : • 7 bits, pas de parité, 1 bit d'arrêt • 8 bits, parité paire, 2 bits d'arrêt • 8 bits, parité impaire, 2 bits d'arrêt
8	Cliquez sur le bouton **Avancé** pour définir les paramètres avancés.

Configuration de la table d'émission/ réception du mode ASCII

La taille maximale des trames émises et/ou reçues s'élève à 256 octets. La table de mots associée à l'instruction EXCHx se compose des tables de contrôle d'émission et de réception.

	Octet de poids fort	**Octet de poids faible**
Table de contrôle	Commande	Longueur (émission/réception)
	Réservés (0)	Réservés (0)
Table d'émission	Octet 1 émis	Octet 2 émis
	...	...
		Octet n émis
	Octet n+1 émis	
Table de réception	Octet 1 reçu	Octet 2 reçu
	...	...
		Octet p reçu
	Octet p+1 reçu	

Table de contrôle

L'octet **Longueur** contient la longueur de la table d'émission en octets (250 max.), qui est écrasée par le nombre de caractères reçus à la fin de la réception, si la réception est demandée.

L'octet **Commande** doit contenir l'un des éléments suivants :

- 0 : Emission seule
- 1 : Emission/réception
- 2 : Réception seule

Tables d'émission/ réception

En mode Emission seule, les tables de contrôle et d'émission sont renseignées avant l'exécution de l'instruction EXCHx ; elles peuvent être de type %KW ou %MW. Aucun espace n'est requis pour la réception des caractères en mode Emission seule. Une fois que tous les octets ont été émis %MSGx.D est mis à 1 ; il est alors possible d'exécuter une nouvelle instruction EXCHx.

En mode Emission/Réception, les tables de contrôle et d'émission sont renseignées avant l'exécution de l'instruction EXCHx ; elles doivent être de type %MW. Un espace prévu pour un maximum de 256 octets de réception est requis à la fin de la table d'émission. Une fois que tous les octets ont été émis, l'automate Twido passe en mode de réception et est prêt à recevoir des octets.

En mode Réception seule, la table de contrôle est renseignée avant l'exécution de l'instruction EXCHx ; elle doit être de type %MW. Un espace prévu pour un maximum de 256 octets de réception est requis à la fin de la table de contrôle. L'automate Twido passe immédiatement en mode de réception et est prêt à recevoir des octets.

La réception est terminée une fois que les octets de fin de trame utilisés ont été reçus ou lorsque la table de réception est pleine. Dans ce cas, une erreur (débordement de la table de réception) apparaît dans le mot %SW63 et %SW64. Si un timeout différent de zéro est configuré, la réception se termine lorsque ce timeout est écoulé. Si vous sélectionnez une valeur de timeout égale à zéro, il n'existe aucun timeout de réception. Par conséquent, pour arrêter la réception, activez l'entrée %MSGx.R.

Echange de messages

Le langage propose deux services pour la communication :

- **Instruction EXCHx :** pour émettre/recevoir des messages.
- **Bloc fonction %MSGx :** pour contrôler les échanges de messages.

L'automate Twido utilise le protocole configuré pour ce port lors du traitement d'une instruction EXCHx.

Note : Il est possible de configurer chaque port de communication pour différents protocoles ou pour le même protocole. Pour accéder à l'instruction EXCHx ou au bloc fonction %MSGx de chaque port de communication, il suffit d'y ajouter le numéro du port (1 ou 2).

Instruction EXCHx

L'instruction EXCHx permet à l'automate Twido d'envoyer et/ou de recevoir des informations vers/depuis des équipements ASCII. L'utilisateur définit une table de mots (%MWi:L ou %KWi:L) contenant des informations de contrôle, ainsi que les données à envoyer et/ou à recevoir (jusqu'à 256 octets en émission et/ou réception). La description du format de la table de mots a été spécifiée précédemment.

Un échange de messages s'effectue à l'aide de l'instruction EXCHx :

Syntaxe : [EXCHx %MWi:L]

où : x = numéro du port (1 ou 2)

L = nombre de mots dans les tables de mots de commande, d'émission et de réception

L'automate Twido doit terminer l'échange de la première instruction EXCHx avant de pouvoir en lancer une deuxième. Il est nécessaire d'utiliser le bloc fonction %MSGx lors de l'envoi de plusieurs messages.

Le traitement de l'instruction par liste EXCHx se produit immédiatement, en sachant que toutes les émissions sont démarrées sous contrôle d'interruptions (la réception des données est également sous contrôle d'interruptions), ce qui est considéré comme un traitement en arrière-plan.

Bloc fonction %MSGx

L'utilisation du bloc fonction %MSGx est facultative ; elle permet de gérer des échanges de données. Le bloc fonction %MSGx remplit trois fonctions :

- **Vérification des erreurs de communication**
  La recherche d'erreurs permet de vérifier que le paramètre L (longueur de la table de mots) programmé à l'aide de l'instruction EXCHx est suffisamment grand pour contenir la longueur du message à envoyer. Ce paramètre est comparé à la longueur programmée dans l'octet de poids faible du premier mot de la table de mots.
- **Coordination de plusieurs messages**
  Pour garantir la coordination lors de l'envoi de plusieurs messages, le bloc fonction %MSGx fournit les informations requises pour déterminer le moment où l'émission du message précédent est terminée.
- **Emission de messages prioritaires**
  Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

Le bloc fonction %MSGx dispose d'une entrée et de deux sorties associées :

Entrée/Sortie	Définition	Description
R	Entrée RAZ	Mise à 1 : réinitialise la communication ou le bloc (%MSGx.E = 0 et %MSGx.D = 1).
%MSGx.D	Communication terminée	0 : requête en cours. 1 : communication terminée en cas de fin de transmission, de réception du caractère de fin, d'erreur ou de réinitialisation du bloc.
%MSGx.E	Erreur	0 : longueur du message et liaison corrects 1 : en cas de commande erronée, de table configurée de manière incorrecte, de mauvais caractère reçu (vitesse, parité, etc.) ou de saturation de la table de réception

Limitations

Il est important de garder à l'esprit les limitations suivantes :

- La disponibilité et le type du port 2 (voir %SW7) sont contrôlés uniquement lors de la mise sous tension ou de la réinitialisation.
- Tout message en cours de traitement sur le port 1 est abandonné lorsque TwidoSuite est connecté.
- Il est impossible de traiter EXCHx ou %MSG sur un port configuré en tant que liaison distante.
- EXCHx abandonne le traitement Modbus esclave actif.
- Le traitement des instructions EXCHx ne fait pas l'objet d'une nouvelle tentative en cas d'erreur.
- Il est possible d'utiliser l'entrée RAZ pour annuler le traitement de la réception d'une instruction EXCHx.
- Il est possible de configurer des instructions EXCHx avec un timeout d'annulation de réception.
- Les messages multiples sont contrôlés via %MSGx.D.

Erreurs et conditions de fonctionnement

Si une erreur se produit lors de l'utilisation de l'instruction EXCHx, les bits %MSGx.D et %MSGx.E sont mis à 1, le mot système %SW63 contient le code d'erreur du port 1 et %SW64 le code d'erreur du port 2.

Mots système	Utilisation
%SW63	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - timeout écoulé 6 - erreur d'émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage de réception plus important que la table de réception 13 - interruption du traitement EXCH par l'automate
%SW64	Code d'erreur EXCH2 : Voir %SW63.

Conséquence du redémarrage de l'automate sur la communication

Lorsqu'un automate redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'automate arrête toutes les communications ASCII.

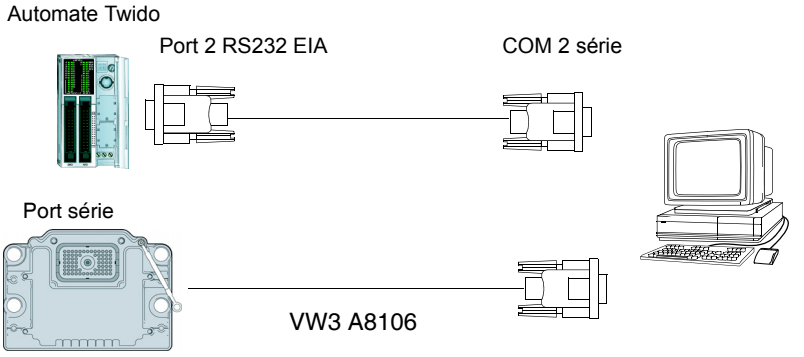
Exemple de liaison ASCII

Pour configurer une liaison ASCII, procédez comme suit :

1. Configurez le matériel.
2. Connectez le câble de communication ASCII.
3. Configurez le port.
4. Ecrivez une application.
5. Initialisez l'éditeur de tables d'animation.

L'illustration suivante représente l'utilisation de la communication ASCII à l'aide d'un émulateur de terminal sur un PC.

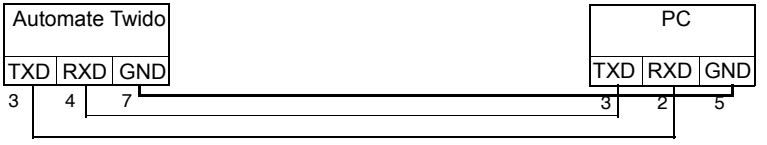
Etape 1 : Configuration du matériel :



La configuration matérielle comporte deux connexions série entre le PC et un automate Twido doté d'un port 2 EIA RS232 en option. Sur un automate modulaire, le port 2 en option correspond à TWDNOZ232D ou à TWDNAC232D dans le TWDXCPODM. Sur l'automate compact, le port 2 optionnel est un port TWDNAC232D. L'automate Twido Extreme TWDLEDCK1 ne possède qu'un port série et, par conséquent, ne possède pas de port 2.

Pour configurer l'automate, connectez le câble TSX PCX1031 (non illustré) au port 1 de l'automate Twido. Connectez ensuite le câble au port COM 1 du PC. Vérifiez que le commutateur est en position 2. Enfin, connectez le port COM 2 du PC au port 2 EIA RS232 de l'automate Twido. Le schéma de câblage est présenté à l'étape suivante.

Etape 2 : Schéma de câblage de communication ASCII (EIA RS232) :



Le nombre minimum de fils utilisé dans un câble de communication ASCII est 3. Croisez les signaux d'émission et de réception.

Note : A l'extrémité PC du câble, des connexions supplémentaires (telles que DTR et DSR) peuvent être nécessaires afin de satisfaire le protocole de transmission. Aucune connexion supplémentaire n'est requise pour l'automate Twido.

Etape 3 : Configuration du port:

Matériel -> Ajouter une option TWDNOZ232D

Port série 2	
Protocole	ASCII
Repère	
Débit	19 200
Bits de données	8
Parité	Aucune
Bit d'arrêt	1
Délai de réponse (x 100 ms)	100
Délai entre les trames (ms)	
Caractère de début	
1er caractère de fin	65
2ème caractère de fin	
Arrêt sur silence (ms)	
Arrêt sur le nombre d'octets reçus	

Emulateur de terminal sur un PC	
Port :	COM2
Débit :	19 200
Données :	8 bits
Parité :	Aucune
Arrêt :	1 bit
Contrôle de flux :	Aucun

Utilisez une simple application d'émulateur de terminal sur le PC pour configurer le port COM2 et garantir l'absence de contrôle de flux.

Utilisez TwidoSuite pour configurer le port de l'automate. En premier lieu, configurez l'option matérielle. Dans cet exemple, le port TWDNOZ232D est ajouté à la base automate modulaire.

En second lieu, initialisez le paramétrage de la communication de l'automate à l'aide des mêmes paramètres que ceux de l'émulateur de terminal sur le PC. Dans cet exemple, la lettre majuscule "A" est choisie comme "premier caractère de fin", afin de terminer la réception de caractère. Un timeout de dix secondes est choisi pour le paramètre "Délai de réponse". Un seul de ces deux paramètres sera utilisé, selon celui qui se produira en premier.

Etape 4 : Ecriture d'une application :

```
LD 1
[%MW10 := 16#0104 ]
[%MW11 := 16#0000 ]
[%MW12 := 16#4F4B ]
[%MW13 := 16#0A0D ]
LD 1
AND %MSG2.D
[EXCH2 %MW10:8]
LD %MSG2.E
ST %Q0.0
END
```

Utilisez TwidoSuite pour créer un programme d'application en trois temps. Tout d'abord, initialisez la table de contrôle et la table d'émission pour utiliser l'instruction EXCH. Dans cet exemple, une commande est configurée pour à la fois envoyer et recevoir des données. La quantité de données à envoyer est réglée sur quatre octets, comme défini dans l'application, suivi du caractère de fin de trame utilisé (dans ce cas, le premier caractère de fin "A"). Les caractères de début et de fin ne sont pas affichés dans la table d'animation où seuls des caractères de données apparaissent. Quoi qu'il en soit, ces caractères sont automatiquement transmis ou vérifiés lors de la réception (par %SW63 et %SW64), lorsqu'ils sont utilisés.

Vérifiez ensuite le bit d'état de communication associé à %MSG2 et exécutez l'instruction EXCH2 uniquement si le port est prêt. Une valeur de 8 mots est spécifiée pour l'instruction EXCH2. Il existe deux mots de commande (%MW10 et %MW11), deux mots à utiliser pour les informations d'émission (%MW12 et %MW13) et quatre mots pour recevoir des données (%MW14 à %MW16).

Finalement, l'état d'erreur du mot %MSG2 est détecté et stocké sur le premier bit de sortie des E/S de la base automate locale. Vous pouvez également effectuer à l'aide de %SW64 une recherche d'erreurs supplémentaire pour rendre celle-ci plus précise.

Etape 5 : Initialisation de l'éditeur de tables d'animation :

Repère	Courant	Format mémorisé
1 %MW10	0104	Hexadécimal
2 %MW11	0000	Hexadécimal
3 %MW12	4F4B	Hexadécimal
4 %MW13	0A0D	Hexadécimal
5 %MW14	TW	ASCII
6 %MW15	ID	ASCII
7 %MW16	O	ASCII

L'étape finale consiste à décharger cette application d'automate et à l'exécuter. Initialisez l'éditeur de tables d'animation pour animer et afficher les mots %MW10 à %MW16. Sur l'émulateur de terminal, les caractères "O - K - CR - LF - A" peuvent s'afficher autant de fois que le délai de réponse du bloc EXCH s'est écoulé. Sur l'émulateur de terminal, tapez "T - W - I - D - O - A". Ces informations sont échangées avec l'automate Twido et s'affichent dans l'éditeur de tables d'animation.

Communications Modbus

Introduction

Le protocole Modbus est un protocole maître-esclave qui permet à un seul et unique maître de demander des réponses à des esclaves ou d'agir en fonction de la requête. Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les réponses aux requêtes de diffusion générale du maître ne sont pas renvoyées.

⚠ ATTENTION

FONCTIONNEMENT INATTENDU DU MATERIEL

- Assurez-vous qu'il existe un seul automate maître Modbus sur le bus et que chaque esclave Modbus dispose d'un repère unique. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.
- Assurez-vous que tous les esclaves Modbus disposent d'un repère unique. Deux esclaves ne doivent pas avoir le même repère. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Configuration matérielle

Il est possible d'établir une liaison Modbus sur le port EIA RS232 ou EIA RS485 et de l'exécuter simultanément sur deux ports de communication au maximum. Chaque port peut obtenir son propre repère Modbus, en utilisant le bit système %S101 et les mots système %SW101 et %SW102 (voir *Bits système (%S), p. 674*).

Le tableau suivant répertorie les équipements qui peuvent être utilisés :

Equipement	Port	Caractéristiques
TWDLC•A10/16/24DRF, TWDLC••40DRF, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Base automate prenant en charge un port EIA RS485 à 3 fils avec un connecteur mini DIN.
TWDLEDCK1	1	Base automate équipée d'un port type EIA RS485 non isolé, longueur maximale limitée à 200 m. **Remarque** : Les options de configuration suivantes ne sont **pas possibles**. ● 7 bits, pas de parité, 1 bit d'arrêt ● 8 bits, parité paire, 2 bits d'arrêt ● 8 bits, parité impaire, 2 bits d'arrêt

Equipement	Port	Caractéristiques
TWDNOZ232D	2	Module de communication équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNAC232D	2	Adaptateur de communication équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur mini DIN. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS485 à 3 fils avec un connecteur de borne. **Remarque :** Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS232 à 3 fils avec un connecteur mini DIN, d'un port EIA RS485 à 3 fils avec un connecteur mini DIN et d'un port EIA RS485 à 3 fils avec un bornier. **Remarque :** Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Communication.

Note : La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate.

Câblage nominal Les connexions de câble nominal sont représentées ci-dessous pour les types EIA RS232 et EIA RS485.

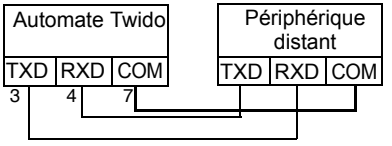
Note : Si le port 1 est utilisé sur l'automate Twido, le signal DPT sur la broche 5 doit être relié au circuit commun (COM) sur la broche 7. Ce signal permet d'indiquer à l'automate Twido que les communications via le port 1 relèvent du protocole Modbus et non du protocole utilisé pour communiquer avec le logiciel TwidoSuite.

Note : Pour l'automate Twido Extreme TWDLEDCK1, si Modbus est utilisé pour la programmation, le contact de la sangle de communication (broche 22) doit être déconnecté. Si la tension appliquée à ce contact (broche 22) est de 0 V, cela indique à l'automate Twido que la communication via le port 1 n'est pas le protocole utilisé pour communiquer avec le logiciel TwidoSuite.

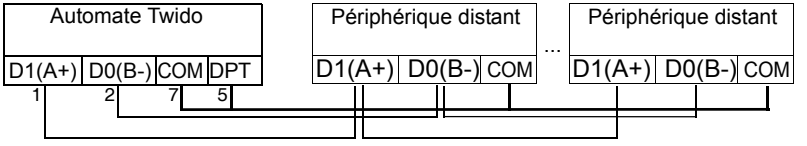
Les connexions de câbles effectuées à chaque équipement distant sont représentées ci-dessous.

Connexion mini DIN

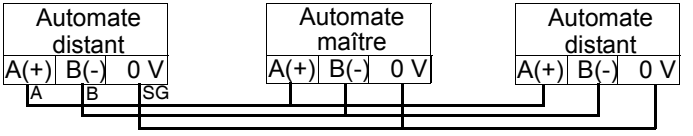
Câble RS232 EIA



Câble RS485 EIA



Connexion bornier



Configuration logicielle

Pour configurer l'automate afin d'utiliser une liaison série pour envoyer et recevoir des caractères à l'aide du protocole Modbus, procédez comme suit :

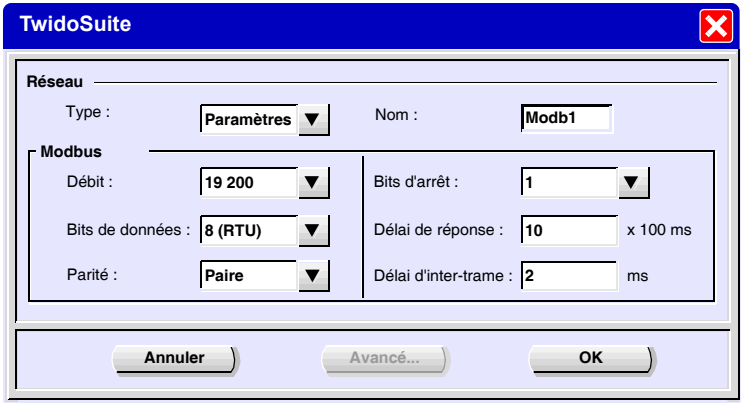
Etape	Description
1	Configurez le port série pour le protocole Modbus à l'aide de TwidoSuite.[1]
2	Créez dans votre application une table d'émission/réception qui sera utilisée par l'instruction EXCHx.

[1]Les options de configuration suivantes ne sont pas possibles pour l'automate Twido Extreme TWDLEDCK1 :

- **7 bits, pas de parité, 1 bit d'arrêt**
- **8 bits, parité paire, 2 bits d'arrêt**
- **8 bits, parité impaire, 2 bits d'arrêt**

Configuration du port

Un automate Twido peut utiliser son port 1 principal ou un port 2 configuré en option pour utiliser le protocole Modbus (l'automate Twido Extreme TWDLEDCK1 ne possède qu'un port série). Pour configurer un port série pour Modbus :

Etape	Action
1	Définissez tous les modules ou adaptateurs de communication supplémentaires configurés dans la base.
2	Déclarez le réseau Modbus dans l'étape Description de TwidoSuite (reportez-vous à la section (Comment créer un réseau (exemple Modbus), Twido, Logiciel de programmation TwidoSuite, Aide en ligne)).
3	Sélectionnez le **port 1** (ou le **port 2** s'il est installé) pour configurer dans la fenêtre Description (reportez-vous à la section (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
4	Pour configurer l'élément Modbus, utilisez l'une des deux méthodes suivantes : ● cliquez sur l'icône Configurer dans la barre d'outils, puis sélectionnez l'élément Modbus dans le graphique descriptif ; ● cliquez deux fois sur l'élément Modbus dans le graphique descriptif.
5	Utilisez l'une des deux méthodes suivantes pour ouvrir la boîte de dialogue Caractéristique (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) des paramètres matériels de liaison Modbus : ● cliquez sur l'icône Configurer dans la barre d'outils, puis sélectionnez la liaison Modbus dans le graphique descriptif ; ● cliquez deux fois sur la liaison Modbus dans le graphique descriptif.
6	Configurez la boîte de dialogue Caractéristique qui s'affiche, en procédant comme suit : 
7	Dans la liste déroulante **Type** de la section **Protocole**, sélectionnez **Modbus**.
8	Définissez les paramètres de communication associés. Les options de configuration suivantes ne sont **pas possibles** pour l'automate Twido Extreme TWDLEDCK1 : ● 7 bits, pas de parité, 1 bit d'arrêt ● 8 bits, parité paire, 2 bits d'arrêt ● 8 bits, parité impaire, 2 bits d'arrêt

Maître Modbus

Le mode Modbus maître permet à l'automate d'envoyer une requête Modbus à un esclave et d'attendre la réponse. Le mode Modbus maître n'est pris en charge que par l'intermédiaire de l'instruction EXCHx. Les modes Modbus ASCII et RTU sont tous les deux pris en charge en mode Modbus maître.

La taille maximale des trames émises et/ou reçues s'élève à 250 octets. En outre, la table de mots associée à l'instruction EXCHx se compose des tables de contrôle, d'émission et de réception.

	Octet de poids fort	**Octet de poids faible**
Table de contrôle	Commande	Longueur (Emission/ Réception)
	Décalage réception	Décalage émission
Table d'émission	Octet 1 émis	Octet 2 émis
	...	...
	...	Octet n émis
	Octet n+1 émis	
Table de réception	Octet 1 reçu	Octet 2 reçu
	...	...
	...	Octet p reçu
	Octet p+1 reçu	

Note : Outre les requêtes faites à chaque esclave, l'automate maître Modbus peut lancer une requête de **diffusion** à tous les esclaves. L'octet **Commande**, dans le cas d'une requête de diffusion générale, doit être réglé sur 00, alors que le **repère esclave** doit être réglé sur 0.

Table de contrôle

L'octet **Longueur** contient la longueur de la table d'émission (250 octets maximum), qui est écrasée par le nombre de caractères reçus à la fin de la réception, si la réception est demandée.

Ce paramètre correspond à la longueur en octets de la table d'émission. Si le paramètre de décalage de l'émission est égal à zéro, il sera égal à la longueur de la trame d'émission. Si le paramètre de décalage de l'émission n'est pas égal à zéro, un octet de la table d'émission (indiqué par la valeur de décalage) ne sera pas émis et ce paramètre sera égal à la longueur de la trame plus 1.

L'octet **Commande** doit toujours être égal à 1 (émission et réception) en cas de requête Modbus RTU (sauf pour une diffusion générale).

L'octet **Décalage émission** contient le rang (1 pour le premier octet, 2 pour le deuxième octet, etc.) dans la table d'émission de l'octet à ignorer lors de l'émission des octets. Il est utilisé pour prendre en charge les émissions associées aux valeurs octet/mot dans le cadre du protocole Modbus. Par exemple, si cet octet est égal à 3, le troisième octet est ignoré, ce qui fait du quatrième octet de la table le troisième octet à émettre.

L'octet **Décalage réception** contient le rang (1 pour le premier octet, 2 pour le deuxième octet, etc.) dans la table de réception à ajouter lors de l'émission des octets. Il est utilisé pour prendre en charge les émissions associées aux valeurs octet/mot dans le cadre du protocole Modbus. Par exemple, si cet octet est égal à 3, le troisième octet de la table est renseigné par un ZERO et le troisième octet réellement reçu est entré dans le quatrième emplacement de la table.

Tables d'émission/ réception

Dans l'un ou l'autre des modes (Modbus ASCII ou Modbus RTU), la table d'émission est écrite avec le contenu de la requête avant l'exécution de l'instruction EXCHx. Au moment de l'exécution, l'automate détermine quelle est la couche liaison de données et effectue toutes les conversions nécessaires pour traiter l'émission et la réponse. Les caractères de début, de fin et de contrôle ne sont pas stockés dans les tables d'émission/réception.

Une fois que tous les octets ont été émis, l'automate passe en mode de réception et est prêt à recevoir des octets.

La réception se termine de l'une des manières suivantes :

- un timeout a été détecté sur un caractère ou une trame ;
- le caractère de fin de trame est reçu en mode ASCII ;
- la table de réception est saturée.

Les entrées **Octet émis X** contiennent les données (codage RTU) de protocole Modbus à émettre. Si le port de communication est configuré en Modbus ASCII, les caractères de trame corrects sont ajoutés à l'émission. Le premier octet comprend le repère de l'équipement (spécifique ou général), le deuxième octet comprend le code de fonction et le reste comprend les informations associées à ce code de fonction.

Note : Il s'agit d'une application type, mais toutes les possibilités ne sont pas définies. Aucune validation des données en cours d'émission n'est effectuée.

Les **Octets reçus X** contiennent les données (codage RTU) de protocole Modbus à recevoir. Si le port de communication est configuré en Modbus ASCII, les caractères de trame corrects sont supprimés de la réponse. Le premier octet comprend le repère de l'équipement, le deuxième octet comprend le code de fonction (ou code de réponse) et le reste comprend les informations associées à ce code de fonction.

Note : Il s'agit d'une application type, mais toutes les possibilités ne sont pas définies. Aucune validation des données en cours de réception n'est effectuée, à l'exception d'une vérification de checksum.

Esclave Modbus

Le mode Modbus esclave permet à l'automate de répondre à des requêtes Modbus standard provenant d'un maître Modbus.

Lorsque le câble TSX PCX1031 est raccordé à l'automate, les communications avec TwidoSuite démarrent sur le port, ce qui désactive temporairement le mode de communication qui était en cours d'exécution avant la connexion de ce câble.

Le protocole Modbus prend en charge deux formats de couche liaison de données : ASCII et RTU. Chaque format est défini par l'implémentation de la couche physique ; le format ASCII utilise sept bits de données tandis que le format RTU en utilise huit.

En mode Modbus ASCII, chaque octet d'un message est envoyé sous la forme de deux caractères ASCII. La trame Modbus ASCII commence par un caractère de début (':') et peut se terminer par deux caractères de fin (CR et LF). Le caractère de fin de trame par défaut est 0x0A (LF). L'utilisateur peut modifier la valeur de cet octet au cours de la configuration. La valeur de contrôle de la trame Modbus ASCII correspond à un simple complément à deux de la trame, excluant les caractères de début et de fin.

Le mode Modbus RTU ne reformate pas le message avant de l'émettre ; cependant, il utilise un mode de calcul de checksum différent, spécifié sous forme de CRC.

Les limitations de la couche liaison de données Modbus sont les suivantes :

- Repère 1-247
- Bits : 128 bits sur demande
- Mots : 125 mots de 16 bits sur demande

Echange de messages

Le langage propose deux services pour la communication :

- **Instruction EXCHx :** pour émettre/recevoir des messages.
- **Bloc fonction %MSGx :** pour contrôler les échanges de messages.

L'automate Twido utilise le protocole configuré pour ce port lors du traitement d'une instruction EXCHx.

Note : Il est possible de configurer chaque port de communication pour différents protocoles ou pour le même protocole. Pour accéder à l'instruction EXCHx ou au bloc fonction %MSGx de chaque port de communication, il suffit d'y ajouter le numéro du port (1 ou 2).

Instruction EXCHx

L'instruction EXCHx permet à l'automate Twido d'envoyer et/ou de recevoir des informations vers/depuis des équipements Modbus. L'utilisateur définit une table de mots (%MWi:L) contenant des informations de contrôle, ainsi que les données à envoyer et/ou à recevoir (jusqu'à 250 octets dans l'émission et/ou réception). La description du format de la table de mots a été spécifiée précédemment.

Un échange de messages s'effectue à l'aide de l'instruction EXCHx :

Syntaxe : [EXCHx %MWi:L]

où : x = numéro du port (1 ou 2)

L = nombre de mots dans les tables de mots de commande, d'émission et de réception

L'automate Twido doit terminer l'échange de la première instruction EXCHx avant de pouvoir en lancer une deuxième. Il est nécessaire d'utiliser le bloc fonction %MSGx lors de l'envoi de plusieurs messages.

Le traitement de l'instruction par liste EXCHx se produit immédiatement, en sachant que toutes les émissions sont démarrées sous contrôle d'interruptions (la réception des données est également sous contrôle d'interruptions), ce qui est considéré comme un traitement en arrière-plan.

Bloc fonction %MSGx

L'utilisation du bloc fonction %MSGx est facultative ; elle permet de gérer des échanges de données. Le bloc fonction %MSGx remplit trois fonctions :

- **Vérification des erreurs de communication**
  La recherche d'erreurs permet de vérifier que le paramètre L (longueur de la table de mots) programmé à l'aide de l'instruction EXCHx est suffisamment grand pour contenir la longueur du message à envoyer. Ce paramètre est comparé à la longueur programmée dans l'octet de poids faible du premier mot de la table de mots.
- **Coordination de plusieurs messages**
  Pour garantir la coordination lors de l'envoi de plusieurs messages, le bloc fonction %MSGx fournit les informations requises pour déterminer le moment où l'émission du message précédent est terminée.
- **Emission de messages prioritaires**
  Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

Le bloc fonction %MSGx dispose d'une entrée et de deux sorties associées :

Entrée/Sortie	Définition	Description
R	Entrée RAZ	Mise à 1 : réinitialise la communication ou le bloc (%MSGx.E = 0 et %MSGx.D = 1)
%MSGx.D	Communication terminée	0 : requête en cours 1 : communication terminée en cas de fin d'émission, de réception du caractère de fin, d'erreur ou de réinitialisation du bloc
%MSGx.E	Erreur	0 : longueur du message et liaison corrects 1 : en cas de commande erronée, de table configurée de manière incorrecte, de mauvais caractère reçu (vitesse, parité, etc.) ou de saturation de la table de réception

Limitations

Il est important de garder à l'esprit les limitations suivantes :

- La présence et la configuration (RS232 ou RS485) du port 2 sont contrôlées lors de la mise sous tension ou de la réinitialisation.
- Tout message en cours de traitement sur le port 1 est abandonné lorsque TwidoSuite est connecté.
- Il est impossible de traiter EXCHx ou %MSG sur un port configuré en tant que liaison distante.
- EXCHx abandonne le traitement Modbus esclave actif.
- Le traitement des instructions EXCHx ne fait pas l'objet d'une nouvelle tentative en cas d'erreur.
- Il est possible d'utiliser l'entrée RAZ pour annuler le traitement de la réception d'une instruction EXCHx.
- Il est possible de configurer des instructions EXCHx avec un timeout d'annulation de réception.
- Les messages multiples sont contrôlés via %MSGx.D.

Erreurs et conditions de fonctionnement

Si une erreur se produit lors de l'utilisation de l'instruction EXCHx, les bits %MSGx.D et %MSGx.E sont mis à 1, le mot système %SW63 contient le code d'erreur du port 1 et %SW64 le code d'erreur du port 2.

Mots système	Utilisation
%SW63	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - timeout écoulé 6 - émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage de réception plus important que la table de réception 13 - interruption du traitement EXCH par l'automate
%SW64	Code d'erreur EXCH2 : voir %SW63.

Redémarrage de l'automate maître

Lorsqu'un automate maître/esclave redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'automate arrête toutes les communications Modbus.

Exemple 1 de liaison Modbus

Pour configurer une liaison Modbus, procédez comme suit :

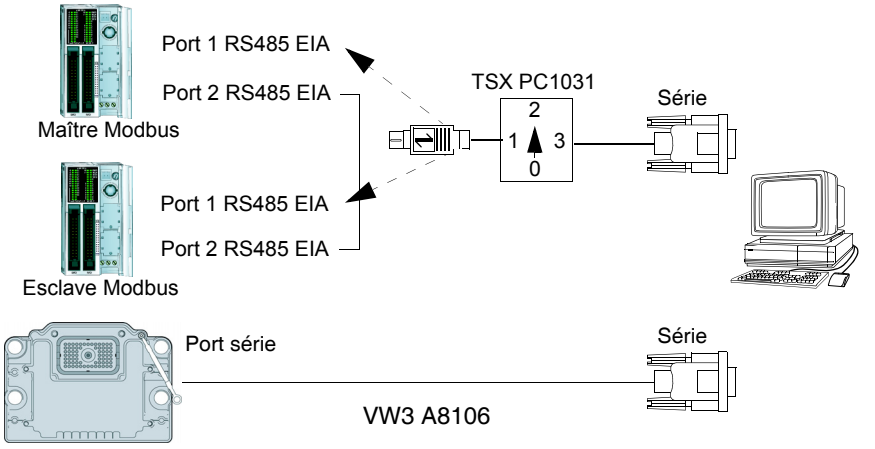
1. Configurez le matériel.[1]
2. Connectez le câble de communication Modbus.
3. Configurez le port.
4. Ecrivez une application.
5. Initialisez l'éditeur de tables d'animation.

Les illustrations suivantes représentent l'utilisation de la requête Modbus 3 pour lire des mots de sortie d'un esclave. Cet exemple utilise deux automates Twido.

[1]Les options de configuration suivantes ne sont **pas possibles** pour l'automate Twido Extreme TWDLEDCK1 :

- 7 bits, pas de parité, 1 bit d'arrêt
- 8 bits, parité paire, 2 bits d'arrêt
- 8 bits, parité impaire, 2 bits d'arrêt

Etape 1 : Configuration du matériel :



La configuration matérielle comprend deux automates Twido. L'un d'entre eux est configuré en tant que Modbus maître et l'autre en tant que Modbus esclave.

Note : Dans cet exemple, chaque automate est configuré afin d'utiliser EIA RS485 sur le port 1 ainsi que EIA RS485 sur le port 2 facultatif. Sur un automate modulaire, le port 2 facultatif peut être de type TWDNOZ485D ou TWDNOZ485T ou, si vous utilisez TWDXCPODM, il peut être de type TWDNAC485D ou TWDNAC485T. Sur un automate compact, le port 2 facultatif peut être un port TWDNAC485D ou TWDNAC485T. L'automate Twido Extreme TWDLEDCK1 ne possède qu'un port série et, par conséquent, ne possède pas de port 2.

Pour configurer chaque automate, connectez le câble TSX PCX1031 au port 1 de l'automate.

Note : Le câble TSX PCX1031 peut uniquement être connecté à un automate à la fois et seulement sur le port 1 EIA RS485.

Connectez ensuite le câble au port COM 1 du PC. Assurez-vous que le commutateur du câble est en position 2. Téléchargez et contrôlez l'application. Répétez cette procédure pour le deuxième automate.

Etape 2 : Connexion du câble de communication Modbus :

Connexion mini DIN

Connexion bornier

Le câblage utilisé dans cet exemple correspond à une simple connexion point à point. Les trois signaux D1(A+), D0(B-) et COM(0V) sont câblés conformément à l'illustration.

En cas d'utilisation du port 1 de l'automate Twido, le signal DPT (broche 5) doit être relié au circuit commun (broche 7). Cette condition du DPT détermine si TwidoSuite est connecté. Lorsqu'il est relié à la terre, l'automate utilise la configuration de port définie dans l'application pour déterminer le type de communication.

Pour l'automate Twido Extreme TWDLEDCK1, si Modbus est utilisé pour la programmation, le contact de la sangle de communication (broche 22) doit être déconnecté. Si la tension appliquée à ce contact (broche 22) est de 0 V, cela indique à l'automate Twido que la communication via le port 1 n'est pas le protocole utilisé pour communiquer avec le logiciel TwidoSuite.

Etape 3 : Configuration du port[1] :

Matériel -> Ajouter une option TWDNOZ485-

Matériel => Paramètre Comm. de l'automate	
Port série 2	
Protocole	Modbus
Repère	1
Débit	19 200
Bits de données	8 (RTU)
Parité	Aucune
Bit d'arrêt	1
Délai de réponse (x 100 ms)	10
Délai entre les trames (ms)	10

Matériel -> Ajouter une option TWDNOZ485-

Matériel => Paramètre Comm. de l'automate	
Port série 2	
Protocole	Modbus
Repère	2
Débit	19 200
Bits de données	8 (RTU)
Parité	Aucune
Bit d'arrêt	1
Délai de réponse (x 100 ms)	100
Délai entre les trames (ms)	10

[1]Les options de configuration suivantes ne sont **pas possibles** pour l'automate Twido Extreme TWDLEDCK1 :

- 7 bits, pas de parité, 1 bit d'arrêt
- 8 bits, parité paire, 2 bits d'arrêt
- 8 bits, parité impaire, 2 bits d'arrêt

Dans les applications maître et esclave, les ports EIA RS485 optionnels sont configurés. Assurez-vous que les paramètres de communication de l'automate sont modifiés en protocole Modbus et à des repères différents.

Dans cet exemple, le maître est réglé sur un repère 1 et l'esclave sur 2. Le nombre de bits est réglé sur 8, ce qui indique que le mode Modbus RTU sera utilisé. S'il avait été de 7, le mode Modbus ASCII aurait été utilisé. La seule autre valeur par défaut modifiée concerne l'augmentation du délai de réponse à 1 seconde.

Note : Etant donné que le mode Modbus RTU a été sélectionné, le paramètre "Fin de trame" a été ignoré.

Etape 4 : Ecriture d'une application :

```
LD 1
[%MW0 := 16#0106]
[%MW1 := 16#0300]
[%MW2 := 16#0203]
[%MW3 := 16#0000]
[%MW4 := 16#0004]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW0 := 16#6566]
[%MW1 := 16#6768]
[%MW2 := 16#6970]
[%MW3 := 16#7172]
END
```

A l'aide de TwidoSuite, un programme d'application est écrit pour le maître et l'esclave. Pour l'esclave, il suffit de définir certains mots mémoire sur un ensemble de valeurs connues. Dans le maître, la table de mots de l'instruction EXCHx est initialisée afin de lire quatre mots de l'esclave au repère Modbus 2 qui démarre à l'emplacement %MW0.

Note : Remarquez l'utilisation du décalage réception défini dans %MW1 du maître Modbus. Le décalage de trois ajoute un octet (valeur = 0) à la troisième position de la zone de réception de la table. Il permet d'aligner les mots dans le maître de façon à ce qu'ils entrent correctement dans les limites de mot. Sans ce décalage, chaque mot de données serait fractionné en deux mots dans le bloc d'échange. Ce décalage est utilisé pour des raisons de commodité.

Avant d'exécuter l'instruction EXCH2, l'application vérifie le bit de communication associé à %MSG2. Enfin, l'état d'erreur du %MSG2 est détecté et stocké sur le premier bit de sortie sur l'E/S de la base automate locale. Il est également possible d'ajouter une recherche d'erreurs supplémentaire à l'aide de %SW64 pour rendre celle-ci plus précise.

Etape 5 : Initialisation de l'éditeur de tables d'animation dans le maître :

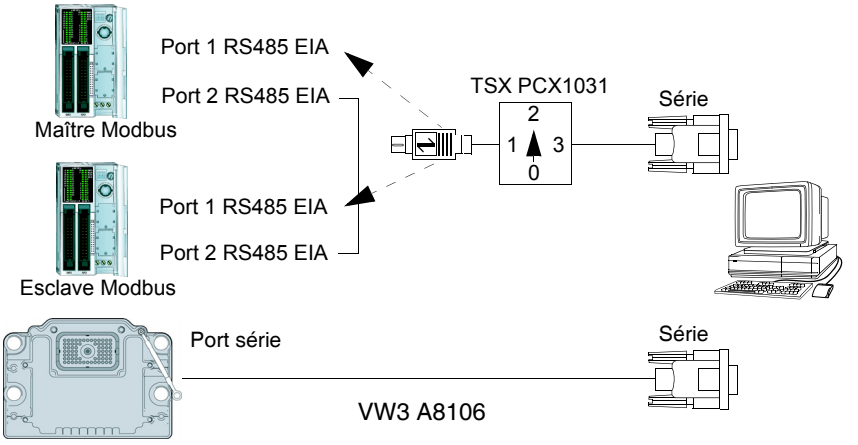
Repère	Courant	Mémorisé	Format
1 %MW5	0203	0000	Hexadécimal
2 %MW6	0008	0000	Hexadécimal
3 %MW7	6566	0000	Hexadécimal
4 %MW8	6768	0000	Hexadécimal
5 %MW9	6970	0000	Hexadécimal
6 %MW10	7172	0000	Hexadécimal

Après le déchargement et la configuration de tous les automates en vue de leur exécution, ouvrez une table d'animation sur le maître. Examinez la section réponse de la table pour vérifier que le code de réponse correspond à 3 et que le nombre d'octets lus est correct. Notez également, dans cet exemple, que les mots lus de l'esclave (commençant par %MW7) sont correctement alignés avec les limites de mot dans le maître.

Exemple 2 de liaison Modbus

L'illustration suivante représente l'utilisation de la requête Modbus 16 pour écrire des mots de sortie sur un esclave. Cet exemple utilise deux automates Twido.

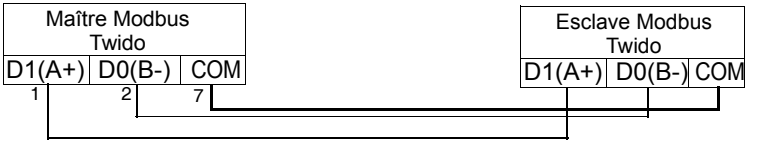
Etape 1 : Configuration du matériel



La configuration matérielle est identique à celle de l'exemple précédent.

Etape 2 : Connexion du câble de communication Modbus (RS485) :

Connexion mini DIN



Connexion bornier

Le câblage de communication Modbus est identique à celui de l'exemple précédent.

Etape 3 : Configuration du port:

Matériel -> Ajouter une option TWDNOZ485-

Matériel => Paramètre Comm. de l'automate	
Port série 2	
Protocole	Modbus
Repère	1
Débit	19 200
Bits de données	8 (RTU)
Parité	Aucune
Bit d'arrêt	1
Délai de réponse (x 100 ms)	10
Délai entre les trames (ms)	10

Matériel -> Ajouter une option TWDNOZ485-

Matériel => Paramètre Comm. de l'automate	
Port série 2	
Protocole	Modbus
Repère	2
Débit	19 200
Bits de données	8 (RTU)
Parité	Aucune
Bit d'arrêt	1
Délai de réponse (x 100 ms)	100
Délai entre les trames (ms)	10

Les configurations du port sont identiques à celles de l'exemple précédent.

Etape 4 : Ecriture d'une application :

```
LD 1
[%MW0 := 16#010C]
[%MW1 := 16#0007]
[%MW2 := 16#0210]
[%MW3 := 16#0010]
[%MW4 := 16#0002]
[%MW5 := 16#0004]
[%MW6 := 16#6566]
[%MW7 := 16#6768]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW18 := 16#FFFF]
END
```

A l'aide de TwidoSuite, un programme d'application est créé pour le maître et l'esclave. Pour l'esclave, écrivez un seul mot mémoire %MW18. Cette action permet d'allouer de l'espace sur l'esclave pour les repères mémoire de %MW0 à %MW18. Sans allocation d'espace, la requête Modbus essaie d'écrire à des emplacements inexistants sur l'esclave.

Dans le maître, la table de mots de l'instruction EXCH2 est initialisée afin d'écrire 4 octets vers l'esclave d'adresse Modbus 2 au repère %MW16 (10 hexadécimal).

Note : Remarquez l'utilisation du décalage émission défini dans %MW1 de l'application du maître Modbus. Le décalage de sept permet de supprimer l'octet de poids fort dans le sixième mot (valeur 00 hexadécimale dans %MW5). Cette action permet d'aligner les valeurs de données dans la table d'émission de la table de mots de façon à ce qu'elles entrent correctement dans les limites de mot.

Avant d'exécuter l'instruction EXCH2, l'application vérifie le bit de communication associé à %MSG2. Enfin, l'état d'erreur du %MSG2 est détecté et stocké sur le premier bit de sortie sur l'E/S de la base automate locale. Vous pouvez également effectuer à l'aide de %SW64 une recherche d'erreurs supplémentaire pour rendre celle-ci plus précise.

Etape 5 : Initialisation de l'éditeur de tables d'animation :

Création de la table d'animation suivante dans le maître :

Repère	Courant	Mémorisé	Format
1 %MW0	010C	0000	Hexadécimal
2 %MW1	0007	0000	Hexadécimal
3 %MW2	0210	0000	Hexadécimal
4 %MW3	0010	0000	Hexadécimal
5 %MW4	0002	0000	Hexadécimal
6 %MW5	0004	0000	Hexadécimal
7 %MW6	6566	0000	Hexadécimal
8 %MW7	6768	0000	Hexadécimal
9 %MW8	0210	0000	Hexadécimal
10 %MW9	0010	0000	Hexadécimal
11 %MW10	0004	0000	Hexadécimal

Création de la table d'animation suivante dans l'esclave :

Repère	Courant	Mémorisé	Format
1 %MW16	6566	0000	Hexadécimal
2 %MW17	6768	0000	Hexadécimal

Après le déchargement et la configuration de tous les automates en vue de la mise en RUN, ouvrez une table d'animation sur l'automate esclave. Les deux valeurs de %MW16 et %MW17 sont écrites sur l'esclave. Dans le maître, il est possible d'utiliser la table d'animation afin d'examiner la partie table de réception des données d'échange. Ces données affichent le repère de l'esclave, le code de réponse, le premier mot écrit et le nombre de mots écrits à partir de %MW8 dans l'exemple ci-dessus.

Requêtes Modbus standard

Introduction

Ces requêtes permettent d'échanger des mots ou bits mémoire entre les périphériques. Le format de table utilisé est le même pour le mode RTU et pour le mode ASCII.

Format	Référence
Bit	%Mi
Mot	%MWi

Maître Modbus : Lecture de N bits

Le tableau suivant représente les requêtes 01 et 02.

	Index de la table	Octet de poids fort	Octet de poids faible
Table de contrôle	0	01 (Emission/Réception)	06 (Longueur émission) (*)
	1	03 (Décalage réception)	00 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	01 ou 02 (Code de requête)
	3	Repère du premier bit à lire	
	4	N_1 = Nombre de bits à lire	
Table de réception (après réponse)	5	Esclave@(1..247)	01 ou 02 (Code de réponse)
	6	00 (octet ajouté à la suite d'une action de Décalage réception)	N_2 = Nombre d'octets des données à lire = [1+(N_1-1)/8], où [] signifie partie intégrale
	7	Valeur du 1er octet (valeur = 00 ou 01)	Valeur du 2ème octet (si N_1>1)
	8	Valeur du 3ème octet (si N_1>1)	
	...		
	(N_2/2) + 6 (si N_2 est pair) (N_2/2+1) + 6 (si N_2 est impair)	Valeur du N_2^{ème} octet (si N_1>1)	

(*) Cet octet reçoit également la longueur de la chaîne émise après réponse

Maître Modbus : Lecture de N mots

Le tableau suivant représente les requêtes 03 et 04.

	Index de la table	**Octet de poids fort**	**Octet de poids faible**
Table de contrôle	0	01 (Emission/Réception)	06 (Longueur émission) (*)
	1	03 (Décalage réception)	00 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	03 ou 04 (Code de requête)
	3	Repère du premier mot à lire	
	4	N = Nombre de mots à lire	
Table de réception (après réponse)	5	Esclave@(1..247)	03 ou 04 (Code de réponse)
	6	00 (octet ajouté à la suite d'une action de Décalage réception)	2*N (nombre d'octets lus)
	7	Premier mot lu	
	8	Deuxième mot lu (si N>1)	
	...		
	N+6	Nième mot lu (si N>2)	

(*) Cet octet reçoit également la longueur de la chaîne émise après réponse

Note : L'opération Décalage réception = 3 ajoute un octet (valeur = 0) à la troisième position de la table de réception, ce qui assure un bon positionnement dans la table, du nombre d'octets lus et des valeurs des mots lus.

Maître Modbus : Ecriture d'un bit

Le tableau suivant représente la requête 05.

	Index de la table	Octet de poids fort	Octet de poids faible
Table de contrôle	0	01 (Emission/Réception)	06 (Longueur émission) (*)
	1	00 (Décalage réception)	00 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	05 (Code de requête)
	3	Repère du bit à écrire	
	4	Valeur du bit à écrire	
Table de réception (après réponse)	5	Esclave@(1..247)	05 (Code de réponse)
	6	Repère du bit écrit	
	7	Valeur écrite	

(*) Cet octet reçoit également la longueur de la chaîne émise après réponse

Note :
- Il n'est pas nécessaire d'utiliser le décalage pour cette requête.
- La trame de la réponse est identique à celle de cette requête (dans un cas normal).
- Pour affecter la valeur 1 à un bit, le mot associé dans la table d'émission doit contenir la valeur FF00H, et 0 pour affecter la valeur 0 à un bit.

Maître Modbus : Ecriture d'un mot

Le tableau suivant représente la requête 06.

	Index de la table	**Octet de poids fort**	**Octet de poids faible**
Table de contrôle	0	01 (Emission/Réception)	06 (Longueur émission) (*)
	1	00 (Décalage réception)	00 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	06 (Code de requête)
	3	Repère du mot à écrire	
	4	Valeur du mot à écrire	
Table de réception (après réponse)	5	Esclave@(1..247)	06 (Code de réponse)
	6	Repère du mot écrit	
	7	Valeur écrite	

(*) Cet octet reçoit également la longueur de la chaîne émise après réponse

Note :
- Il n'est pas nécessaire d'utiliser le décalage pour cette requête.
- La trame de la réponse est identique à celle de cette requête (dans un cas normal).

Maître Modbus : Ecriture de N bits

Le tableau suivant représente la requête 15.

	Index de la table	**Octet de poids fort**	**Octet de poids faible**
Table de contrôle	0	01 (Emission/Réception)	8 + nombre d'octets (émission)
	1	00 (Décalage réception)	07 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	15 (Code de requête)
	3	Numéro du premier bit à écrire	
	4	N_1 = Nombre de bits à écrire	
	5	00 (octet non envoyé, effet de décalage)	N_2 = Nombre d'octets des données à écrire = [1+(N_1-1)/8], où [] signifie partie intégrale
	6	Valeur du 1^{er} octet	Valeur du $2^{ème}$ octet
	7	Valeur du $3^{ème}$ octet	Valeur du $4^{ème}$ octet
	...		
	(N_2/2) + 5 (si N_2 est pair) (N_2/2+1) + 5 (si N_2 est impair)	Valeur du $N_2^{ème}$ octet	
Table de réception (après réponse)		Esclave@(1..247)	15 (Code de réponse)
		Repère du 1^{er} bit écrit	
		Repère des bits écrits (= N_1)	

Note :

- L'opération Décalage émission = 7 supprime le 7ème octet de la trame envoyée. Elle permet également d'assurer une bonne correspondance entre les valeurs des mots de la table d'émission.

Maître Modbus : Ecriture de N mots

Le tableau suivant représente la requête 16.

	Index de la table	**Octet de poids fort**	**Octet de poids faible**
Table de contrôle	0	01 (Emission/Réception)	8 + (2*N) (Longueur émission)
	1	00 (Décalage réception)	07 (Décalage émission)
Table d'émission	2	Esclave@(1..247)	16 (Code de requête)
	3	Repère du premier mot à écrire	
	4	N = Nombre de mots à écrire	
	5	00 (octet non envoyé, effet de décalage)	2*N = Nb d'octets à écrire
	6	Première valeur du mot à écrire	
	7	Deuxième valeur à écrire	
	...		
	N+5	N valeurs à écrire	
Table de réception (après réponse)	N+6	Esclave@(1..247)	16 (Code de réponse)
	N+7	Repère du premier mot écrit	
	N+8	Repère des mots écrits (= N)	

Note : L'opération Décalage émission = 7 supprime le 5ème octet MMSB de la trame envoyée. Elle permet également d'assurer une bonne correspondance entre les valeurs des mots de la table d'émission.

Classe d'implémentation Transparent Ready (Twido série A05, Ethernet A15)

Vue d'ensemble

Les codes de fonction Modbus suivants sont pris en charge par le protocole Modbus série et le protocole Modbus TCP/IP. Pour plus d'informations sur le protocole Modbus, reportez-vous au document *Protocole d'application Modbus* disponible à l'adresse http://www.modbus-ida.org

Codes de fonction Modbus pris en charge par Twido (MB FC)

Le tableau suivant décrit les codes de fonction pris en charge par le protocole série Twido et le protocole Modbus TCP/IP :

Supported MB FC	Code Sub-fc pris en charge	Fonction
1	—	Lire plusieurs bits internes %M
2	—	Lire plusieurs bits internes %M
3	—	Lire plusieurs registres internes %MW
4	—	Lire plusieurs registres internes %MW
5	—	Forcer un seul bit interne %M
6	—	Ecrire un seul registre interne %MW
8	00 uniquement	Diagnostic de l'écho
15	—	Ecrire plusieurs bits internes %M
16	—	Ecrire plusieurs registres internes %MW
23	—	Lire/Ecrire plusieurs registres internes %MW
43	14	Lire l'identification de l'équipement (service normal)

Fonctions analogiques intégrées

6

Présentation

Objet de ce chapitre

Cette rubrique décrit la gestion de la voie analogique et des potentiomètres analogiques intégrés.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Point de réglage analogique	150
Voie analogique	152

Point de réglage analogique

Introduction

Les automates Twido possèdent :

- un point de réglage analogique sur les automates TWDLC•A10DRF et TWDLC•A16DRF et sur tous les automates modulaires (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK et TWDLMDA40DUK),
- deux points de réglage sur les automates TWDLC•A42DRF et TWDLC••40DRF.

Programmation

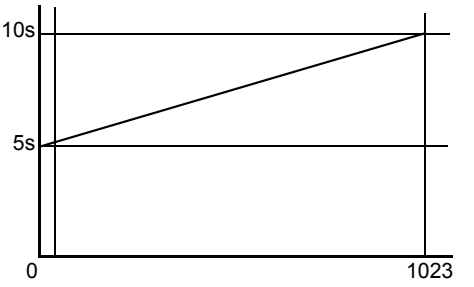
Les valeurs numériques, allant de 0 à 1 023 pour le point de réglage analogique 1 et de 0 à 511 pour le point de réglage analogique 2, correspondant aux valeurs analogiques données par ces points de réglage analogiques sont contenues dans les deux mots d'entrée suivants :

- %IW0.0.0 pour le point de réglage analogique 1 (situé à gauche)
- %IW0.0.1 pour le point de réglage analogique 2 (situé à droite)

Ces mots peuvent être utilisés dans les opérations arithmétiques et pour n'importe quel type de réglage (présélection d'une temporisation ou d'un compteur, ajustement de la fréquence du générateur d'impulsions ou de la durée de préchauffage d'une machine, etc.).

Exemple

Utilisation du point de réglage analogique 1 pour modifier la durée de temporisation de 5 à 10 secondes :

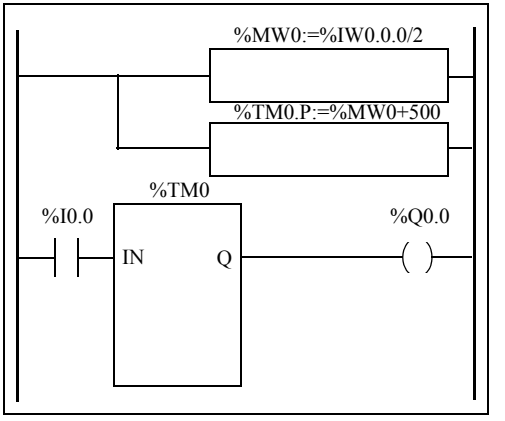
Ce réglage utilise la quasi-totalité de la plage du point de réglage analogique 1 (0 à 1 023).	

Les paramètres suivants sont sélectionnés au moment de la configuration du bloc de temporisation %TM0 :

- Type TON
- Base temps : 10 ms

La valeur de présélection de la durée de temporisation est calculée à partir de la valeur du point de réglage analogique, à l'aide de l'équation suivante : %TM0.P := (%IW0.0.0/2)+500.

Code pour l'exemple précédent :




```
LD      1
[%MW0:=%IW0.0.0/2]
[%TM0.P:=%MW0+500]
BLK     %TM0
LD      %I0.0
IN
OUT_BLK
LD      Q
ST      %Q0.0
END_BLK
.....................
```

Voie analogique

Introduction

Tous les automates modulaires (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK et TWDLMDA40DUK) possèdent une voie analogique. La tension en entrée est comprise entre 0 et 10 V et entre 0 et 511 pour le signal numérisé. La voie analogique utilise un schéma de calcul de moyennes simple qui s'applique sur huit échantillons.

Principe

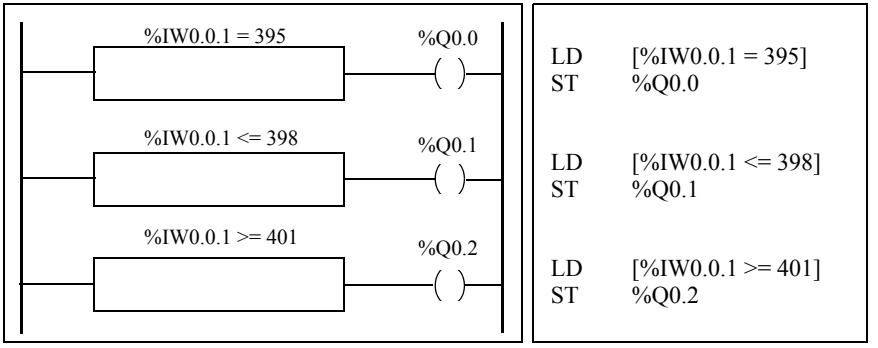
Un convertisseur de données analogiques en données numériques échantillonne une tension allant de 0 à 10 V en une valeur numérique allant de 0 à 511. Cette valeur est stockée dans le mot système %IW0.0.1. La valeur est linéaire sur l'intégralité de la plage, et chaque incrément correspond à 20 mV (10 V/512) environ. Dés que le système détecte la valeur 511, la voie est considérée comme saturée.

Exemple de programmation

Régulation de la température d'un four : La température de cuisson est réglée sur 350 °C. Une variation de +/- 2,5 °C engendre une disjonction des sorties %Q0.0 et %Q0.2. La quasi-totalité de la plage de paramètres possibles de la voie analogique (de 0 à 511) est utilisée dans cet exemple. Les paramètres analogiques des différentes températures sont les suivants :

Température (°C)	Tension	Mot système %IW0.0.1
0	0	0
347,5	7,72	395
350	7,77	398
352,5	7,83	401
450	10	511

Code pour l'exemple précédent :



Gestion des modules analogiques

7

Présentation

Objet de ce chapitre

Ce chapitre offre une présentation des procédures de gestion des modules analogiques des automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation des modules analogiques	154
Adressage d'entrées et de sorties analogiques	155
Configuration d'entrées et de sorties analogiques	156
Informations sur l'état du module analogique	163
Exemples d'utilisation de modules analogiques	165

Vue d'ensemble des modules analogiques

Introduction

Outre le point de réglage analogique 10 bits et la voie analogique 9 bits, l'ensemble des automates Twido prenant en charge l'expansion d'E/S sont également capables de communiquer avec des modules d'E/S analogiques.

Ces modules analogiques sont les suivants :

Nom	Voies	Plage du signal	Codage
TWDAMI2HT	2 en entrée	0 à 10 V ou 4 à 20 mA	12 bits
TWDAMI2LT	2 en entrée	Entrées - du thermocouple	bit 16
TWDAMO1HT	1 en sortie	0 à 10 V ou 4 à 20 mA	12 bits
TWDAMM3HT	2 en entrée, 1 en sortie	0 à 10 V ou 4 à 20 mA	12 bits
TWDAMM6HT	4 en entrée, 2 en sortie	0 à 10 V ou 4 à 20 mA	12 bits
TWDALM3LT	2 en entrée, 1 en sortie	0 à 10 V, Entrées Th ou PT100, Sorties de 4 à 20 mA	12 bits
TWDAVO2HT	2 en sortie	+/- 10 V	11 bits signe +
TWDAMI4LT	4 en entrée	0 à 10 V, 0 à 20 mA, capteurs à 3 fils NI ou PT	12 bits
TWDAMI8HT	8 en entrée	0 à 10 V ou 0 à 20 mA	10 bits
TWDARI8HT	8 en entrée	Capteurs NTC ou PTC	10 bits

Fonctionnement des modules analogiques

Les mots en entrée et en sortie (%IW et %QW) sont utilisés pour échanger des données entre l'application utilisateur et les voies analogiques. La mise à jour de ces mots est effectuée de manière synchronisée avec la scrutation de l'automate en mode RUN.

⚠ ATTENTION

DÉMARRAGE INATTENDU DES ÉQUIPEMENTS

Lorsque l'automate est en position STOP, la sortie analogique se trouve en position de repli. Dans le cas d'une sortie TOR, la consigne par défaut est zéro.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

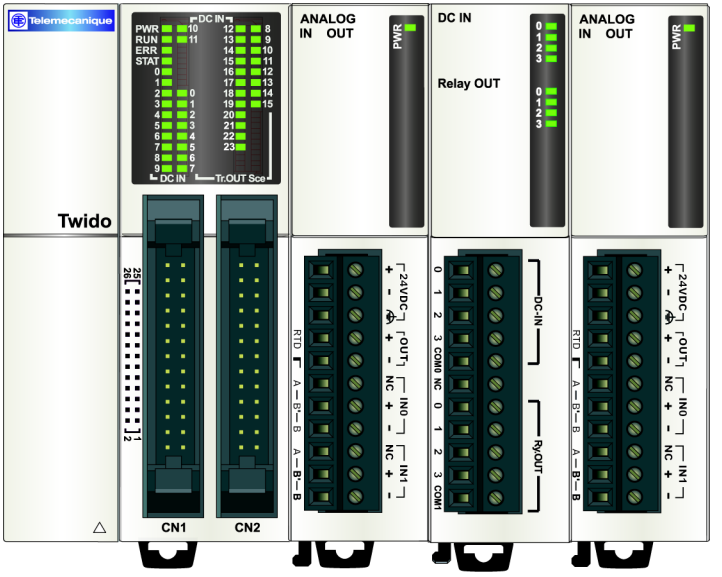
Adressage d'entrées et de sorties analogiques

Introduction

Des adresses sont affectées aux voies analogiques en fonction de leur emplacement sur le bus d'expansion.

Exemple d'adressage d'E/S analogique

Dans cet exemple, un module TWDLMDA40DUK possède un potentiomètre analogique 10 bits intégré, ainsi qu'une voie analogique 9 bits intégrée. Sur le bus d'expansion, sont configurés : un module analogique TWDAMM3HT, un module de relais numérique d'E/S TWDDMM8DRT, ainsi qu'un second module analogique TWDAMM3HT.



Le tableau suivant présente une description détaillée de l'adressage de chaque sortie.

Description	Base	Module 1	Module 2	Module 3
Point de réglage analogique 1	%IW0.0.0			
Voie analogique intégrée	%IW0.0.1			
Voie 1 d'entrée analogique		%IW0.1.0		%IW0.3.0
Voie 2 d'entrée analogique		%IW0.1.1		%IW0.3.1
Voie 1 de sortie analogique		%QW0.1.0		%QW0.3.0
Voies d'entrée numérique			%I0.2.0 - %I0.2.3	
Voies de sortie numérique			%Q0.2.0 -%Q0.2.3	

Configuration d'entrées et de sorties analogiques

Introduction

Cette section présente des informations sur la configuration des entrées et des sorties du module analogique.

Configuration d'E/S analogique

Utilisez l'éditeur de configuration (Editeur de configuration, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) pour régler les paramètres des modules d'E/S analogiques que vous avez ajoutés en tant que modules d'expansion lors de la description du système (voir (Positionnement d'un module, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)).

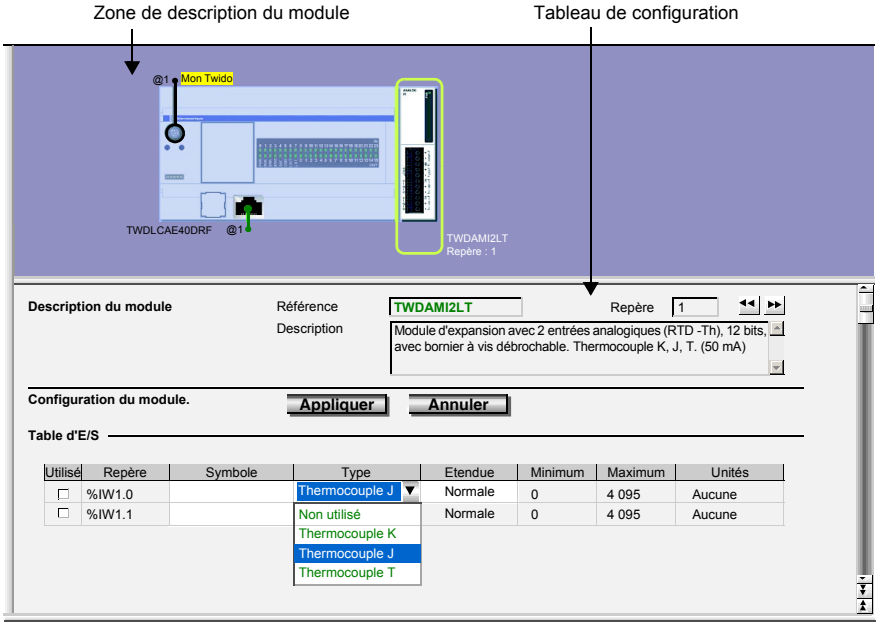
Note : Tous les paramètres de configuration d'E/S peuvent être modifiés en modes connecté et local. Par exemple, pour le module TWDAMI2LT, le type d'entrée (J, K ou T) peut être modifié autant en mode connecté qu'en mode local.

Contenu de l'éditeur de configuration

Dans **Programme** → **Configurer** → **Configuration du matériel**, le volet de configuration affiche une zone **Description** contenant la référence et une brève description du module suivis du tableau de configuration du module.

Si votre système contient plusieurs modules, pour afficher le tableau de configuration du module approprié, cliquez sur le module souhaité dans le volet graphique supérieur.

L'exemple suivant représente le volet de configuration matérielle du module TWDAMI2LT.



Le tableau fournit des indications sur les éléments : **Repère**, **Symbole**, **Type**, **Plage**, **Minimum**, **Maximum** et **Unités**.

- Dans TWDAMI4LT et TWIDAMI8HT, le tableau est précédé d'une zone de liste **Type d'entrée**.
- Dans TWDAVO2HT et TWDAMI8HT, la colonne **Type** est remplacée par une colonne **Configurée** avec des cases à cocher.
- Dans TWDARI8HT, chaque voie (0 à 7) est configurée individuellement dans un onglet dans lequel vous pouvez choisir la méthode de configuration **Graphique** ou **Formule**. Vous pouvez visualiser le tableau dans l'onglet **Recap**.

Description

La zone **Description** décrit brièvement ce module.

Repère

Chaque ligne du tableur représente une voie d'entrée ou de sortie du module.

Les repères sont identifiés dans le tableau ci-dessous, où le "i" représente l'emplacement du module sur le bus d'expansion.

Nom du module	Repère
TWDAMI2LT	**2 entrées (%IWi.0, %IWi.1)**
TWDALM3LT	2 entrées (%IWi.0, %IWi.1), 1 sortie (%QWi.0)
TWDAMM3HT	2 entrées (%IWi.0, %IWi.1), 1 sortie (%QWi.0)
TWDAMM6HT	4 entrées (%IWi.0 à %IWi.3), 2 sorties (%QWi.0, %QWi.1)
TWDAMI2HT	2 entrées (%IWi.0, %IWi.1)
TWDAMO1HT	1 sortie (%QWi.0)
TWDAVO2HT	2 sorties (%QWi.0, %QWi.1)
TWDAMI4LT	4 entrées (%IWi.0 à %IWi.3)
TWDAMI8HT	8 entrées (%IWi.0 à %IWi.7)
TWDARI8HT	8 entrées (%IWi.0 à %IWi.7)

Symbole

Affichage en lecture seule d'un symbole du repère, si ce dernier a été affecté.

Type d'entrée et de sortie

Identifie le mode d'une voie. Le choix dépend de la voie et du type du module.

Pour TWDAMO1HT, TWDAMM3HT et TWDALM3LT, vous pouvez configurer le type de voie de sortie unique comme suit :

Type
Non utilisé
0 - 10 V
4 - 20 mA

Pour TWDAMM6HT, vous pouvez configurer les 4 types de voie d'entrée et les 2 types de voie de sortie comme suit :

Type d'entrée
0 - 10 V
4 - 20 mA

Pour TWDAMI2HT et TWDAMM3HT, vous pouvez configurer les 2 types de voie d'entrée comme suit :

Type
Non utilisé
0 - 10 V
4 - 20 mA

Pour TWDAMI2LT*, vous pouvez configurer les 2 types de voie d'entrée comme suit :

Type
Non utilisé
Thermocouple K
Thermocouple J
Thermocouple T

Note : *Avant d'utiliser le module TWDLAMI2LT, vérifiez que la version du microprogramme de votre automate est la version 4.0 ou ultérieure.

Pour TWDALM3LT, vous pouvez configurer les 2 types de voie d'entrée comme suit :

Type
Non utilisé
Thermocouple K
Thermocouple J
Thermocouple T
PT 100

Pour TWDAVO2HT, aucun réglage de type n'est nécessaire.

Pour TWDAMI4LT, vous pouvez configurer les quatre types d'entrée comme suit :

Type d'entrée	Type
Tension	Non utilisé 0 - 10 V
Courant	Non utilisé 0 - 20 mA
Température	Non utilisé PT 100 PT 1000 NI 100 NI 1000

Pour TWDAMI8HT, vous pouvez configurer les 8 types d'entrée comme suit :

Type d'entrée
0 - 10 V
0 - 20 mA

Pour TWDARI8HT, vous pouvez configurer chaque voie d'entrée (0 à 7) individuellement dans le champ **Fonctionnement** situé dans la partie inférieure de la fenêtre. Choisissez directement un **Mode** et une **Plage**, si nécessaire. Vous pouvez ensuite afficher un résumé de toutes les informations dans l'onglet Recap, avec une colonne **Type** indiquant :

Type
Non utilisé
NTC / CTN
PTC / CTP

⚠ ATTENTION

DÉTÉRIORATION DU MATÉRIEL

Si vous raccordez votre entrée afin de mesurer la tension et que vous configurez TwidoSuite pour le courant, vous risquez d'endommager le module analogique de façon irréversible. Assurez-vous que le raccordement est conforme à la configuration TwidoSuite.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Plage

Identifie la plage de valeurs d'une voie. Les choix dépendent du type spécifique de voie et de module.

Une fois le **Type** configuré, vous pouvez définir la **Plage** correspondante. Un tableau indique les valeurs **Minimum** et **Maximum** acceptées (fixes ou définies par l'utilisateur), ainsi que les **Unités**, le cas échéant.

Plage (Capteurs NTC)	Minimum	Maximum	Unités	Modules analogiques d'E/S
Normale	0	4 095	Aucune	TWDAMI2LT TWDALM3LT TWDAMO1HT TWDAMM3HT TWDAMM6HT TWDAMI2HT TWDAMI4LT
	-2 048	2 047		TWDAVO2HT
	0	1 023		TWDAMI8HT TWDARI8HT
Personnalisé	Défini par l'utilisateur avec un minimum de -32 768	Défini par l'utilisateur avec un maximum de 32 767	Aucune	Tous les modules analogiques d'E/S

Plage (Capteurs NTC)	Minimum	Maximum	Unités	Modules analogiques d'E/S
Celsius	K : -2 700 J : -2 000 T : -2 700	K : 13 700 J : 7 600 T : 4 000	0,1 °C	TWDAMI2LT
	K : 0 J : 0 T : 0	K : 13 000 J : 12 000 T : 40 000		TWDALM3LT
	Mis à jour dynamiquement par TwidoSuite selon les paramètres définis par l'utilisateur			TWDARI8HT
	-2 000	6 000		TWDAMI4LT (Capteur Pt)
	-500	1 500		TWDAMI4LT (Capteur Ni)
Fahrenheit	K : -4 540 J : -3 280 T : -4 540	K : 24 980 J : 14 000 T : 7 520	0,1 °F	TWDAMI2LT
	K : 320 J : 320 T : 320	K : 23 720 J : 21 920 T : 7 520		TWDALM3LT
	Mis à jour dynamiquement par TwidoSuite selon les paramètres définis par l'utilisateur			TWDARI8HT
	-3 280	11 120		TWDAMI4LT (Capteur Pt)
	-580	3 020		TWDAMI4LT (Capteur Ni)
Résistance	100	10 000	Ohm	TWDARI8HT
	74	199		TWDAMI4LT (Ni100)
	742	1987		TWDAMI4LT (Ni1000)
	18	314		TWDAMI4LT (Pt100)
	184	3 138		TWDAMI4LT (Pt1000)

Méthode graphique ou formule

Dans TWDARI8HT, chaque voie (0 à 7) est configurée individuellement dans un onglet. Cochez la case **Utilisé**, puis choisissez entre la méthode de configuration **Graphique** et **Formule**.

- **Méthode Graphique**
  (**R1**, **T1**) et (**R2**, **T2**) correspondent aux coordonnées de deux points appartenant à la courbe, ces coordonnés étant exprimées au format en virgule flottante.
  Les valeurs **R1**(8 700 par défaut) et **R2** (200 par défaut) sont exprimées en ohms.
  L'unité des valeurs **T1** (233,15 par défaut) et **T2** (398,15 par défaut) peut être définie dans la zone de liste **Unités** : **Kelvin** (par défaut), **Celsius** ou **Fahrenheit**.
  Remarque : La modification de l'unité de température après avoir défini les valeurs T1 et T2 n'entraîne pas un recalcul automatique des valeurs T1 et T2 avec la nouvelle unité.
- **Méthode formule**
  Si vous connaissez les paramètres **Rref**, **Tref** et **B**, vous pouvez utiliser cette méthode pour définir les caractéristiques du capteur.
  Rref (330 par défaut) est exprimé en ohms.
  B est égal à 3 569 par défaut (min. 1, max. 32 767).
  L'unité de la valeur **Tref** (298,15 par défaut) peut être définie dans la zone de liste **Unités** : **Kelvin** (par défaut), **Celsius** ou **Fahrenheit**.

Voici un tableau des valeurs **Tref** minimales et maximales correspondantes entre les unités :

Unité	Valeur min.	Valeur max.
Kelvin	1	650
Celsius	-272	376
Fahrenheit	-457	710

Dans les deux fenêtres Graphique et Formule, vous pouvez importer des valeurs depuis une autre voie dans la voie en cours de configuration :

1. Sélectionnez un numéro de voie dans la zone **Voie n°**.
2. Cliquez sur le bouton **Importer valeurs**.

Certains messages d'avertissement ou d'erreurs peuvent être associés à ces fenêtres.

Note : Si vous commencez à définir ces valeurs et que vous décidez de basculer de la méthode Graphique à la méthode Formule ou de la méthode Formule à la méthode Graphique, un message d'avertissement apparaît et indique que les valeurs par défaut seront appliquées et que toute valeur modifiée sera perdue.

Informations sur l'état du module analogique

Tableau d'état Le tableau suivant contient les informations nécessaires pour contrôler l'état des modules d'E/S analogique.

Mot système	Fonction	Description
%SW80	Etat de l'E/S de base	**Pour le module analogique standard, %SW8x est décrit comme suit :** Bit [0] : toutes les voies analogiques en fonctionnement normal Bit [1] : module en phase d'initialisation Bit [2] : alimentation par défaut Bit [3] : configuration par défaut Bit [4] : conversion de la voie d'entrée 0 en cours Bit [5] : conversion de la voie d'entrée 1 en cours Bit [6] : paramètre de la voie d'entrée 0 non valide Bit [7] : paramètre de la voie d'entrée 1 non valide Bit [8 et 9] : non utilisés Bit [10] : dépassement par valeur supérieure de la voie d'entrée 0 Bit [11] : dépassement par valeur supérieure de la voie d'entrée 1 Bit [12] : dépassement par valeur inférieure de la voie d'entrée 0 Bit [13] : dépassement par valeur inférieure de la voie d'entrée 1 Bit [14] : non utilisé Bit [15] : paramètre de la voie de sortie non valide
%SW80	Etat de l'E/S de base	**Pour le module analogique standard TWDAMI4LT, %SW8x est décrit comme suit :** Bit [0 et 1] : état de la voie 0 0 0: Voie analogique en fonctionnement normal 0 1: paramètre de la voie d'entrée non valide 1 0: valeur d'entrée indisponible (module en phase d'initialisation, conversion en cours) 1 1: valeur de la voie d'entrée non valide (dépassement par valeur inférieure ou supérieure) Bit [2 et 3] : état de la voie 1 (même description que le bit [0 et 1]) Bit [4 et 5] : état de la voie 2 (même description que le bit [0 et 1]) Bit [6 et 7] : état de la voie 3 (même description que le bit [0 et 1]) Bit [8 et 15] : non utilisés

Mot système	Fonction	Description
%SW80	Etat de l'E/S de base	**Pour le module analogique standard TWDAMI8HT %SW8x est décrit comme suit :** Bit [0 et 1] : état de la voie 0 0 0: Voie analogique en fonctionnement normal 0 1: paramètre de la voie d'entrée non valide 1 0: valeur d'entrée indisponible (module en phase d'initialisation, conversion en cours) 1 1: valeur de la voie d'entrée non valide (dépassement par valeur inférieure ou supérieure) Bit [2 et 3] : état de la voie 1 (même description que le bit [0 et 1]) Bit [4 et 5] : état de la voie 2 (même description que le bit [0 et 1]) Bit [6 et 7] : état de la voie 3 (même description que le bit [0 et 1]) Bit [8 et 9] : état de la voie 4 (même description que le bit [0 et 1]) Bit [10 et 11] : état de la voie 5 (même description que le bit [0 et 1]) Bit [et 13] : état de la voie 6 (même description que le bit [0 et 1]) Bit [14 et 15] : état de la voie 7 (même description que le bit [0 et 1])
%SW81	Etat du module d'expansion d'E/S 1 : Définitions identiques à %SW80	
%SW82	Module d'expansion d'E/S 2 Etat : Définitions identiques à %SW80	
%SW83	Module d'expansion d'E/S 3 Etat : Définitions identiques à %SW80	
%SW84	Module d'expansion d'E/S 4 Etat : Définitions identiques à %SW80	
%SW85	Module d'expansion d'E/S 5 Etat : Définitions identiques à %SW80	
%SW86	Module d'expansion d'E/S 6 Etat : Définitions identiques à %SW80	
%SW87	Module d'expansion d'E/S 7 Etat : Définitions identiques à %SW80	

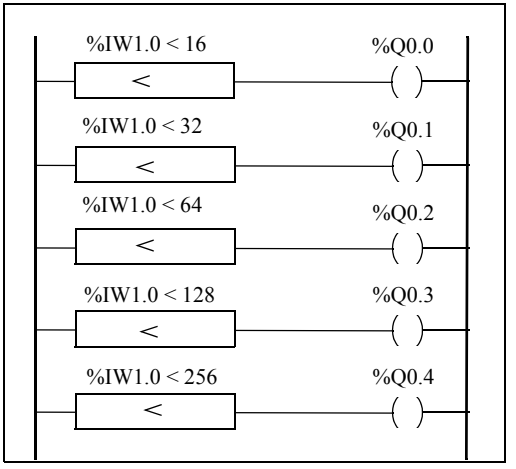
Exemples d'utilisation de modules analogiques

Introduction

Cette section présente un exemple d'utilisation des modules analogiques des automates Twido.

Exemple : entrée analogique

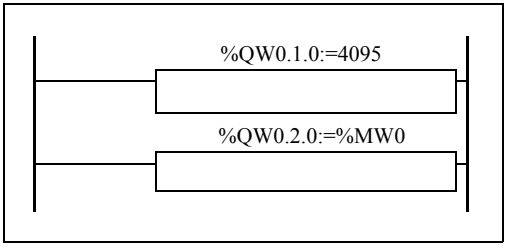
Cet exemple compare le signal d'entrée analogique avec cinq valeurs de seuil distinctes. Une comparaison de l'entrée analogique est effectuée et un bit est réglé sur la base automate si le signal d'entrée est inférieur ou égal au seuil.




```
LD [%IW1.0 < 16]
ST %Q0.0

LD [%IW1.0 < 32]
ST %Q0.1

LD [%IW1.0 < 64]
ST %Q0.2

LD [%IW1.0 < 128]
ST %Q0.3

LD [%IW1.0 < 256]
ST %Q0.4
```

Exemple : sortie analogique

Le programme suivant utilise une carte analogique dans les emplacements 2 et 1. La carte utilisée dans l'emplacement 10 possède une sortie à volts avec une plage "normale" :




```
LD   1
[%QW0.1.0:=4095
LD   1
[%QW0.2.0:=%MW0
```

- Exemple de valeurs de sorties pour %QW1.0=4095 (cas normal) :

Le tableau ci-dessous donne la valeur de la tension de sortie suivant la valeur maximale attribuée à %QW1.0 :

	valeur numérique	**valeur analogique (volt)**
Minimum	0	0
Maximum	4 095	10
Valeur 1	100	0,244
Valeur 2	2 460	6

- Exemple de valeurs de sorties pour une gamme personnalisée (minimum =0, maximum =1000) :

Le tableau ci-dessous donne la valeur de la tension de sortie suivant la valeur maximale attribuée à %QW1.0 :

	valeur numérique	**valeur analogique (volt)**
Minimum	0	0
Maximum	1 000	10
Valeur 1	100	1
Valeur 2	600	6

Configuration d'entrée/sortie Twido Extreme

8

Présentation

Objet de ce chapitre

Ce chapitre propose une vue d'ensemble des entrées et sorties de base Twido Extreme et décrit l'adressage d'E/S.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
8.1	Introduction aux entrées et aux sorties Twido Extreme	168
8.2	Configuration des entrées Twido Extreme	171
8.3	Configuration des sorties de Twido Extreme	190

8.1 Introduction aux entrées et aux sorties Twido Extreme

Adressage des entrées/sorties (E/S) Twido Extreme

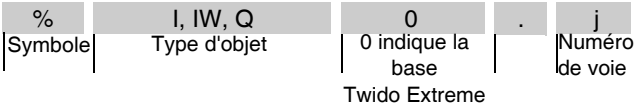
Introduction

La base Twido Extreme possède un connecteur unique à 70 broches qui comprend l'unité d'alimentation, les entrées, les sorties et le bus de communication. Cette section fournit une vue d'ensemble des entrées et sorties et de leurs repères.

Repères d'E/S

Chaque point d'E/S (entrée/sortie) d'une configuration Twido possède un repère unique. Par exemple, le repère "%I0.0.4" est affecté à l'entrée 4 de l'automate.

Le format de repère se présente comme suit :



%I est utilisé pour l'entrée TOR, %IW pour l'entrée (et le PWM) analogique et %Q pour la sortie (et le PWM) TOR.

Types d'entrées/ sorties

Le tableau suivant propose un récapitulatif des types et nombres d'entrées et sorties Twido Extreme et de leurs repères :

ENTREE DU COMMUTATEUR A CLE (1 au total)			
ENTREES (22 au total)			
	Type	Nombre maximal	Plage de repères
TOR	Basculer vers la mise à la terre (source)	11	%I0.0 - %I0.10
	Basculer vers la pile (négatif)	2	%I0.11, %I0.12
Analogique	Capteur analogique actif	4	%IW0.0 - %IW0.3
	Capteur analogique passif	3	%IW0.4 - %IW0.6
Analogique /PWM	Capteur analogique actif ou modulation de largeur[1] (1 kHz maximum)	1	%IW0.7
PWM	Modulation de largeur (5 kHz maximum)	1	%IW0.8
SORTIES (19 au total)			
	Type	Numéro	Repère
TOR	Pilote positif à courant TOR 1 A	1	%Q0.4
	Pilote positif à courant TOR 50 mA, SECU dédié, sortie d'état de l'automate	1	%Q0.3
	Pilote négatif à courant numérique 12/24 V 300 mA	14	%Q0.5 - %Q0.18
PLS/PWM	PLS ou PWM (10 Hz à 1 kHz)	2	%Q0.0, %Q0.1
	PLS ou PWM (10 Hz à 5 kHz)	1	%Q0.2
[1]Remarque : L'entrée %IW0.7 peut être utilisée comme une entrée analogique active ou comme une entrée PWM.			

Pour plus d'informations sur les caractéristiques et l'emplacement de la broche de ces entrées et sorties, reportez-vous à la section (Emplacement des contacts sur le connecteur, Twido Extreme, Guide de référence du matériel).

Commutateur à clé

Le commutateur à clé est une entrée spéciale (physique) qui permet de :

- mettre sous et hors tension le Twido Extreme ;
- placer le Twido Extreme en mode redondant.

Note : Pour permettre le redémarrage A CHAUD à partir du mode redondant, le Twido Extreme ne doit pas être déconnecté de l'unité d'alimentation. Si l'alimentation n'est pas préservée, l'automate réalise un redémarrage A FROID et les données de date et d'heure sont perdues.

En mode redondant, le Twido Extreme préserve l'activité de la RAM et l'intégrité des données de RTC mais cela implique que l'automate doit être connecté à l'unité d'alimentation (car il ne possède pas de pile interne). Pour illustrer davantage cette caractéristique, une analogie au contact d'une voiture peut être envisagée. Le commutateur à clé fonctionne comme le contact d'une voiture qui serait relié à l'autoradio. Lorsque le moteur est éteint, l'autoradio l'est également mais les programmations, l'heure et les autres données sont conservées dans la mémoire. Tant que la radio n'est pas déconnectée de la batterie, elle redémarre avec toutes ses données intactes à chaque démarrage du moteur.

Aucun repère n'a été affecté au commutateur à clé dans le tableau ci-dessus étant donné qu'il s'agit d'un type spécial d'entrée qui ne peut pas être configuré dans TwidoSuite et qui, par conséquent, ne dispose pas de repère dédié.

Mises à jour des E/S

Les bits d'entrée (%I pour les entrées TOR), les mots en entrée (%IW pour les entrées analogiques) et les bits de sortie (%Q) sont utilisés pour échanger des données entre l'application utilisateur et les voies analogiques ou TOR. La mise à jour de ces objets est effectuée de manière synchrone avec la scrutation de l'automate en mode RUN.

8.2 Configuration des entrées Twido Extreme

Présentation

Objet de cette section

Cette section décrit la configuration des entrées pour l'automate Twido Extreme.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration des entrées TOR de Twido Extreme	172
Configuration d'entrée analogique Twido Extreme	178
Configuration d'entrée PWM Twido Extreme	182
Exemple de configuration d'entrée PWM de Twido Extreme	184

Configuration des entrées TOR de Twido Extreme

Introduction

Cette section décrit comment configurer les entrées TOR.

Entrée TOR

Il existe deux types principaux d'entrée TOR :

- Entrée Basculer vers la mise à la terre
- Entrée Basculer vers la pile (+)

Une entrée TOR comprend des valeurs d'entrée, de front montant et de front descendant. Les valeurs de front montant et de front descendant sont calculées à partir des données de l'image en cours et de l'image précédente, issues de deux scrutations consécutives.

Chaque entrée peut être filtrée, forcée ou mémorisée.

Filtrage d'entrée

Les filtrages d'entrées réduisent les bruits sur l'entrée d'un automate. Définir une valeur de 3 ou 12 ms garantit que les modifications soudaines des niveaux d'entrée (dues au bruit) sont ignorées sauf si le nouveau niveau d'entrée persiste pendant 3 ou 12 ms.

Forçage d'entrée

Les valeurs forcées peuvent être affectées aux valeurs d'entrée et de sortie du programme en langage liste d'instructions/schéma à contacts. Ceci peut être utile pour la mise au point. Cette opération est décrite à la section Forçage de valeurs d'entrée/sortie (Forçage de valeurs d'entrée/de sortie, Aide en ligne TwidoSuite).

Mémorisation d'entrée

La mémorisation d'entrée est une fonction spéciale pouvant être affectée à l'une ou à l'ensemble des quatre entrées (%I0.0 à %I0.3) de Twido Extreme. Cette fonction est utilisée pour enregistrer (ou mémoriser) toute impulsion ayant une durée inférieure à celle du temps de scrutation de l'automate. Lorsqu'une impulsion est plus courte qu'une scrutation et que sa valeur est supérieure ou égale à 1 ms, l'automate mémorise l'impulsion qui est ensuite mise à jour à la scrutation suivante. En raison du bruit, une entrée mémorisée doit durer plus d'une ms avant d'être reconnue comme un front montant. Ce mécanisme de mémorisation reconnaît uniquement les fronts montants. Les fronts descendants ne peuvent pas être mémorisés. L'affectation des entrées à mémoriser est effectuée via l'écran de configuration de l'entrée TOR présenté ci-dessous.

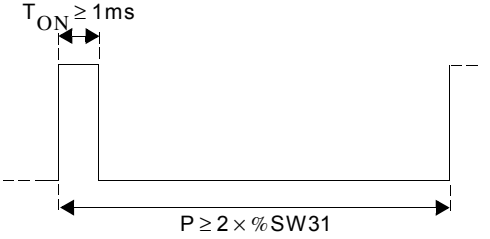
Pour garantir une détection correcte du signal des impulsions lorsque l'option d'entrée à mémorisation d'état est sélectionnée, la largeur d'impulsion (T_{ON}) et la période (P) doivent répondre aux deux exigences suivantes :

- $T_{ON} \geq 1$ ms
- La période du signal d'entrée (P) doit au moins équivaloir à deux fois le temps de scrutation maximum du programme (%SW31) :

$$P \geq 2 \times \%SW31$$

Note : Si cette condition n'est pas remplie, il risque de manquer certaines impulsions.

L'illustration suivante montre les exigences en matière de signal d'entrée lorsque l'entrée à mémorisation d'état est utilisée :



Repères d'entrée TOR

Le tableau suivant présente les repères affectés aux entrées TOR Twido Extreme :

ENTREES TOR (13 au total)		
Type	Nombre max	Plage de repères
Basculer vers la mise à la terre (source)	11	%I0.0 - %I0.10
Basculer vers la pile (négatif)	2	%I0.11, %I0.12

Le commutateur à clé n'est pas inclus dans le tableau ci-dessus étant donné qu'il s'agit d'un type spécial d'entrée qui ne peut pas être configuré dans TwidoSuite et qui ne dispose pas de repère dédié. Pour plus d'informations, reportez-vous à la *p. 168*.

Note : Dans les programmes en langage schéma à contacts, la syntaxe du repère d'E/S est %I0.i (i=0...19) et %Q0.j (j=0...18), telle qu'elle apparaît dans les tableaux de configuration. Cependant, dans les programmes à liste d'instructions, la syntaxe %IW0.0.i et %Q0.0.j est utilisée pour faire référence à ces mêmes repères. Les entrées %I0.13-%I0.19 sont des mappages TOR d'entrées analogiques.

Configuration d'entrée TOR

Les entrées TOR utilisées dans ce programme en langage schéma à contacts/liste d'instructions peuvent être affichées et configurées dans le volet Configuration du module, qui s'affiche lorsque vous sélectionnez **Programme** → **Configurer** → **Configurer le matériel** dans TwidoSuite.

L'onglet Entrée du volet Configuration du module répertorie toutes les entrées utilisées et disponibles, comme illustré ci-dessous :

Utilisé	Repère	Symbole	Utilisé par	Filtrage	Mémorisé ?	Run/Stop ?	Déclenchement	Haute Priorité	Numéro SR
☐	%I0.0			3 ms	☐	☐	Non utilisé	☐	
☐	%I0.1			3 ms	☐	☐	Non utilisé	☐	
☐	%I0.2			3 ms	☐	☐	Non utilisé	☐	
☐	%I0.3			3 ms	☐	☐	Non utilisé	☐	
☐	%I0.4			3 ms	☐	☐			
☐	%I0.5			3 ms	☐	☐			
☐	%I0.6			3 ms	☐	☐			
☐	%I0.7			3 ms	☐	☐			
☐	%I0.8			3 ms	☐	☐			
☐	%I0.9			3 ms	☐	☐			
☐	%I0.10			3 ms	☐	☐			
☐	%I0.11			3 ms	☐	☐			
☐	%I0.12			3 ms	☐	☐			
☐	%I0.13			Aucun filtre	☐	☐			
☐	%I0.14			Aucun filtre	☐	☐			
☐	%I0.15			Aucun filtre	☐	☐			
☐	%I0.16			Aucun filtre	☐	☐			
☐	%I0.17			Aucun filtre	☐	☐			
☐	%I0.18			Aucun filtre	☐	☐			
☐	%I0.19			Aucun filtre	☐	☐			

Note : En mode connecté, les valeurs d'entrée sont également affichées.

Les quatre premières entrées %I0.0 - %I0.3 peuvent être mémorisées ou utilisées pour des événements (front montant, front descendant ou les deux) et ainsi être liées à un sous-programme.

Les treize premières entrées %I0.0 - %I0.12 peuvent être filtrées (3 ms ou 12 ms) et l'une d'elles peut être utilisée pour la fonction RUN/STOP.

%I0.0 à %I0.10 sont des entrées Basculer vers la mise à la terre.

%I0.11 à %I0.13 sont des entrées Basculer vers la pile.

%I0.13 à %I0.19 sont des entrées analogiques liées à une modification d'état de 0 à 1 (ou de 1 à 0) avec des câbles de type Basculer vers la mise à la terre de 3,1 V (ou 1,2 V). Il s'agit d'entrées qui peuvent être mappées à des entrées TOR. Si %IW est utilisée pour ces entrées TOR dans le programme, elles sont considérées comme analogiques et si %I0 est utilisée, elles sont considérées comme des mappages TOR. Ces entrées ne peuvent pas être filtrées, mémorisées ou associées à un événement.

Champs de configuration d'entrée

Les champs de configuration d'entrée affichés dans l'illustration ci-dessus sont détaillés dans le tableau suivant. Comme indiqué, certains champs de ce tableau sont uniquement destinés à l'affichage et ne peuvent pas être modifiés.

Champ	Valeurs possibles	Fonction
Utilisé	Case à cocher activée ou non. Une case cochée indique que l'élément est utilisé. Une case non cochée indique que l'élément n'est pas utilisé.	Uniquement pour l'affichage. Affiche les entrées utilisées par le programme.
Repère	Repères d'entrée %I0.0-%I0.19	Repère de chaque entrée TOR.
Symbole	Valeur définie par l'utilisateur : 32 caractères alphanumériques maximum	Permet de donner un nom à l'entrée. Ce champ peut être modifié. Une fois appliqué, ce nom est alors affiché dans le programme en langage schéma à contacts ou liste d'instructions.
Utilisé par	logique utilisateur, événement	Uniquement pour l'affichage. Répertorie les blocs fonction utilisant cette entrée ou indique si l'entrée est utilisée pour déclencher un événement.
Filtre	Liste déroulante disposant des options suivantes : ● Aucun filtre ● 3 ms ● 12 ms	Les filtres sont utilisés pour réduire les bruits sur l'entrée d'un automate.
Mémorisé ?	Case à cocher activée ou non. Ceci s'applique uniquement aux quatre premières entrées %I0.0 à %I0.3. Les entrées associées à un filtre ne peuvent pas être mémorisées. Les entrées associées à des événements ne peuvent pas être mémorisées. Une case cochée indique que l'élément est mémorisé. Une case non cochée indique que l'élément n'est pas mémorisé.	La mémorisation d'état permet de capturer et d'enregistrer les impulsions entrantes ayant des largeurs d'amplitude inférieures à la durée du cycle de l'automate.
Run/Stop ?	Case à cocher activée ou non. Ceci s'applique uniquement aux treize premières entrées %I0.0 - %I0.12. Une case cochée correspond à RUN. Une case non cochée correspond à STOP.	Pour démarrer ou arrêter le programme d'un automate.
Déclenchement	Liste déroulante proposant les options suivantes : ● Non utilisé ● Front montant ● Front descendant ● Les deux fronts	Pour configurer le déclenchement ou non d'un événement par une entrée.

Champ	Valeurs possibles	Fonction
Haute Priorité	Case à cocher activée ou non. Une seule entrée (%I0.0 à %I0.3) peut être sélectionnée. Il peut uniquement être sélectionné si un événement est déclenché (si le contenu du champ Déclenchement est différent de "Non utilisé").	Pour définir un événement comme étant hautement prioritaire (déclenché).
Numéro SR	Un numéro de sous-programme sélectionné dans la liste déroulante. Il peut uniquement être sélectionné si un événement est déclenché (si le contenu du champ Déclenchement est différent de "Non utilisé").	Le Numéro SR permet d'associer un numéro de sous-programme à un événement (déclenché).

Configuration d'entrée analogique Twido Extreme

Introduction

Cette section décrit la configuration des entrées analogiques et donne un exemple.

Entrée analogique

On distingue trois types d'entrée analogique :

- **Active** (capteur) entrée analogique
- **Passive** (capteur) entrée analogique
- Entrée **PWM**

Les capteurs actifs (Active Analog Sensor Inputs, Twido Extreme Hardware Guide) utilisent une alimentation externe pour les signaux.

Les capteurs passifs (Passive Analog Sensor Inputs, Twido Extreme Hardware Guide) utilisent une partie de l'énergie du signal.

Cette section décrit la configuration des entrées PWM.

Repères d'entrées analogiques

Le tableau suivant présente les repères affectés aux entrées analogiques Twido Extreme :

ENTREES ANALOGIQUES (9 au total)		
Type	Nombre max	Plage de repères
Active	4	%IW0.0 - %IW0.3
Passive	3	%IW0.4 - %IW0.6
Analogique/PWM	1	%IW0.7
PWM	1	%IW0.8

Configuration d'entrées analogiques

Les entrées analogiques utilisées dans ce programme en langage schéma à contacts/ liste d'instructions peuvent être affichées et configurées dans le volet Configuration du module **Programme** → **Configurer** → **Configurer le matériel** de TwidoSuite.

Faites défiler l'onglet Entrée configuration du module qui répertorie les entrées analogiques utilisées et disponibles comme illustré ci-dessous :

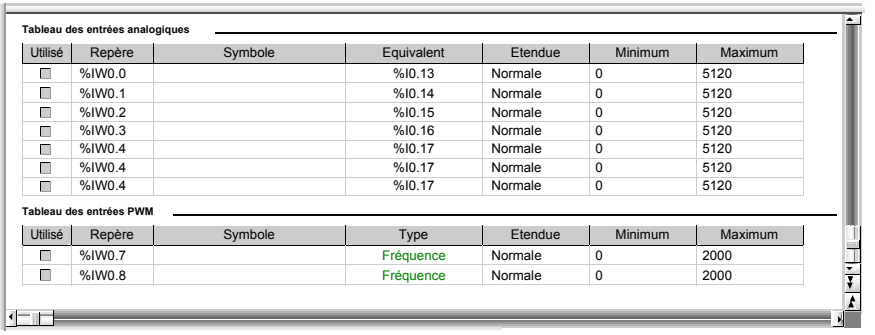


Tableau des entrées analogiques

Utilisé	Repère	Symbole	Equivalent	Etendue	Minimum	Maximum
☐	%IW0.0		%I0.13	Normale	0	5120
☐	%IW0.1		%I0.14	Normale	0	5120
☐	%IW0.2		%I0.15	Normale	0	5120
☐	%IW0.3		%I0.16	Normale	0	5120
☐	%IW0.4		%I0.17	Normale	0	5120
☐	%IW0.4		%I0.17	Normale	0	5120
☐	%IW0.4		%I0.17	Normale	0	5120

Tableau des entrées PWM

Utilisé	Repère	Symbole	Type	Etendue	Minimum	Maximum
☐	%IW0.7		Fréquence	Normale	0	2000
☐	%IW0.8		Fréquence	Normale	0	2000

Note : En mode connecté, les valeurs d'entrée sont également affichées.

Les quatre premières entrées %IW0.0 - %IW0.3 sont des entrées Active.

Les quatre entrées suivantes %IW0.4 - %IW0.6 sont des entrées Passive.

%I0.13 à %I0.19 sont des entrées analogiques (modification de l'état allant de 0 à 1 (ou de 1 à 0) avec des câbles Basculer vers la mise à la terre de 3,1 V (ou 1,2V) pouvant être mappées à des entrées TOR. Si %IW est utilisée pour celles du programme, elles sont considérées comme analogiques et si %I0 est utilisée, elles sont considérées comme des mappages TOR.

%IW0.7 et %IW0.8 sont des entrées PWM et sont décrites à la section suivante. Cependant, %IW0.7 peut également être utilisé pour les entrées analogiques si besoin.

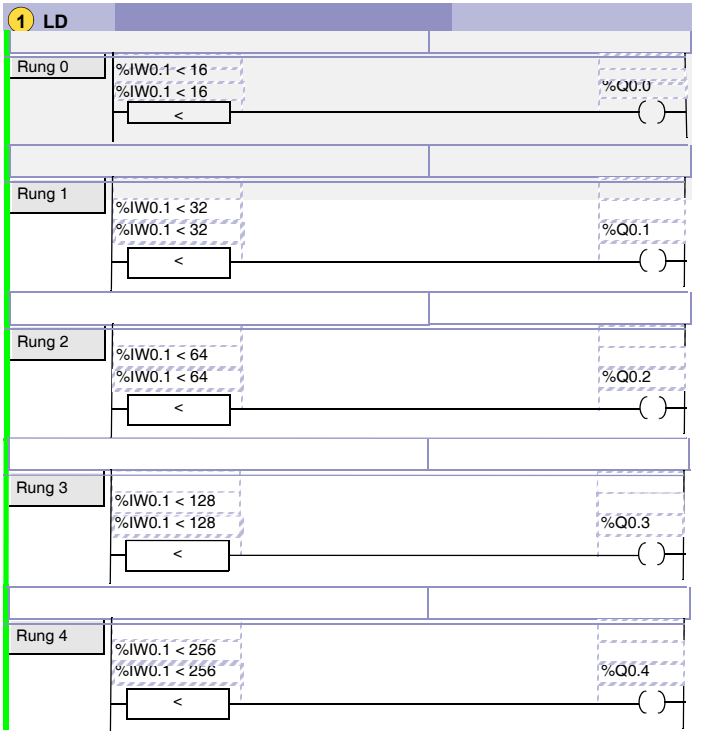
Champs de configuration d'entrée

Les champs de configuration d'entrée affichés dans l'illustration ci-dessus sont détaillés dans le tableau suivant. Comme indiqué, certains champs de ce tableau sont uniquement destinés à l'affichage et ne peuvent pas être modifiés.

Champ	Valeurs possibles	Fonction
Utilisé	Case à cocher activée ou non. Une case cochée indique que l'élément est utilisé. Une case non cochée correspond à un élément inutilisé.	Uniquement pour l'affichage. Affiche les entrées utilisées par le programme. Ce champ ne peut pas être modifié.
Repère	Repères d'entrée %IW0.0-%IW0.6	Repère de chaque entrée analogique.
Symbole	Valeur définie par l'utilisateur : 32 caractères alphanumériques maximum	Permet de donner un nom à l'entrée. Ce champ peut être modifié. Une fois appliqué, ce nom est alors affiché dans le programme en langage schéma à contacts ou liste d'instructions.
Equivalent	%I0.13 à %I0.19	Uniquement pour l'affichage. Répertorie les mappages TOR équivalents.
Etendue	Liste déroulante proposant les options suivantes : ● Normal (par défaut) ● Personnalisés	Permet à la plage de tension d'être sélectionnée (mode normal) ou modifiée (mode personnalisé) dans les champs adéquats.
Maximum/ Minimum	Mode normal : min 0, max 5 120 en mV.	Plage de tension en mV Ces champs ne peuvent pas être modifiés.
	Mode Personnalisé : min -32 768, max +32 767 dans les unités définies par l'utilisateur.	Permet la personnalisation de l'entrée analogique. En mode Personnalisé, ces champs peuvent être modifiés.

Exemple : entrée analogique

Cet exemple compare le signal d'entrée analogique avec cinq valeurs de seuil distinctes. Une comparaison de l'entrée analogique est effectuée et, si elle est inférieure au seuil, un bit est défini dans Twido Extreme.



Même exemple ci-dessous avec un programme liste d'instructions :

```
0 LD [ %IW0.0.1<16 ]
1 ST %Q0.0.0
2 LD [ %IW0.0.1<32 ]
3 ST %Q0.0.1
4 LD [ %IW0.0.1<64 ]
5 ST %Q0.0.2
6 LD [ %IW0.0.1<128 ]
7 ST %Q0.0.3
8 LD [ %IW0.0.0<256 ]
9 ST %Q0.0.4
```

Note : Dans les programmes en langage schéma à contacts, la syntaxe du repère d'E/S est %IW0.i (0=0...8) et %Q0.j (j=0...18) comme affichée dans les tableaux de configuration. Cependant, dans les programmes à liste d'instructions, la syntaxe %IW0.0.i et %Q0.0.j est utilisée pour se référer aux mêmes repères.

Configuration d'entrée PWM Twido Extreme

Vue d'ensemble

Cette section décrit la configuration des entrées PWM.

Entrée PWM

Modulation de largeur (de l'anglais Pulse Width Modulation) : Type spécial d'entrée convertissant une entrée de signal rectangulaire en un paramètre (calculé à partir du signal).

Ce type d'entrée est particulièrement utile lors de la création d'un environnement stable, moins sujet au bruit.

Par exemple, un levier à axe unique (reportez-vous à la *p. 184*) peut être utilisé de manière plus précise et avec moins de risques de dysfonctionnements dus au bruit. Il s'agit d'une caractéristique propre à l'automate Twido Extreme.

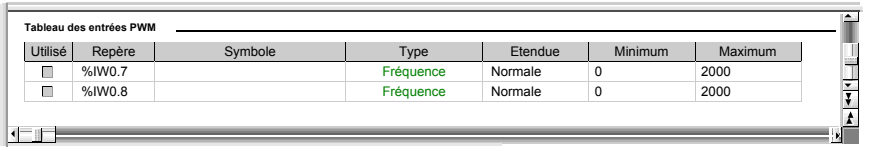
Repères d'entrée PWM

Les repères %IW0.7 et %IW0.8 sont dédiés aux entrées PWM Twido Extreme mais %IW0.7 peut également être utilisé comme entrée analogique.

Configuration des entrées PWM

Les sorties PWM peuvent être utilisées et configurées dans le volet Configuration du module, qui s'affiche lorsque vous sélectionnez **Programme** → **Configurer** → **Configurer le matériel** dans TwidoSuite.

Faites défiler le volet Configuration pour afficher le tableau des entrées PWM comme illustré ci-dessous :



Champs de configuration des entrées

Les champs de configuration de l'entrée PWM sont décrits dans le tableau suivant. Comme indiqué, certains champs de ce tableau sont uniquement destinés à l'affichage (lecture uniquement) et ne peuvent pas être modifiés.

Champ	Valeurs possibles	Fonction
Utilisé	Case à cocher activée ou non. Une case cochée indique que l'élément est utilisé. Une case non cochée indique que l'élément n'est pas utilisé.	Uniquement pour l'affichage. Affiche les entrées utilisées par le programme. Ce champ ne peut pas être modifié.
Repère	Repères d'entrée %IW0.7,%IW0.8	Repère de chaque entrée PWM/analogique.
Symbole	Valeur définie par l'utilisateur : 32 caractères alphanumériques maximum	Permet de donner un nom à l'entrée. Ce champ peut être modifié. Une fois appliqué, ce nom est alors affiché dans le programme en langage schéma à contacts ou liste d'instructions.
Type	Liste déroulante proposant les options suivantes : ● Analogique ● Fréquence (par défaut) ● Rapport ● Largeur d'impulsion	Sélectionnez Analogique si une entrée analogique est utilisée. S'il s'agit d'une entrée PWM, sélectionnez dans quel paramètre vous souhaitez convertir le signal d'entrée PWM.
Etendue	Liste déroulante proposant les options suivantes : ● Normal (par défaut) ● Personnalisé	Permet à la plage d'être personnalisée dans les champs adéquats.
Maximum/ Minimum	Mode normal : Fréquence : min 0, max 20 000 Rapport : min 0, max 100 Largeur d'impulsion : min 0, max 20 000	En mode Normal, ces champs ne peuvent pas être modifiés.
	Mode Personnalisé : Pour tous les types : min -32768, max 32767	Permet à l'entrée analogique/PWM d'être personnalisée. En mode Personnalisé, ces champs peuvent être modifiés.

Exemple de configuration d'entrée PWM de Twido Extreme

Introduction

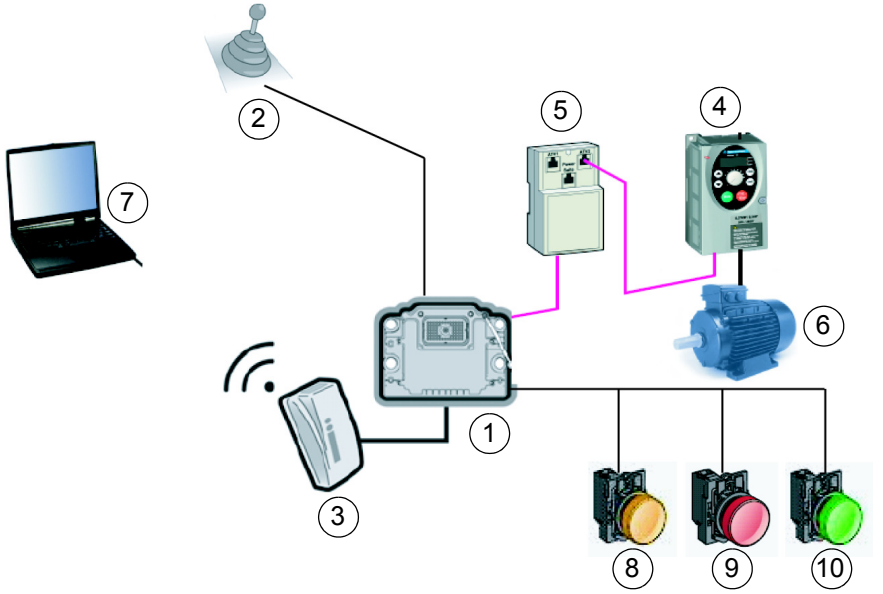
Cette section explique, à l'aide d'un exemple d'application, comment configurer une entrée PWM Twido Extreme.

Dans cet exemple, vous apprendrez à :

- configurer une entrée PWM de manière à accepter un levier à axe unique ;
- utiliser cette entrée pour contrôler la vitesse et la direction d'un moteur.

Exemple d'entrée PWM

Comme illustré ci-dessous, un moteur est contrôlé via le levier à axe unique à l'aide d'une sortie PWM et d'un variateur ATV31. La position du levier fournit la vitesse et assure la direction de la rotation (en avant or en arrière).



Le matériel mentionné dans cet exemple est répertorié ci-dessous :

Numéro sur le schéma	Référence	Description
1	TWDLEDCK1	Automate Twido Extreme
2	TWDXPUJ1A	Levier à axe unique
3	VW3A8114	Verrou électronique BlueTooth : pour permettre une communication sans fil entre le PC et l'automate
4	ATV31H037M2A	Lecteur Altivar ATV31 : pour modifier la vitesse et la direction du moteur
5	VW3CANTAP2	Boîte de raccordement CANopen (TAP) : pour connecter le lecteur ATV31 à l'automate
6	-	Moteur 0,37 kW 1 490 tr/min
7	-	PC avec logiciel TwidoSuite
8	XB6AV5BB	Unité de signalisation jaune Harmony style 6
9	XB6AV4BB	Unité de signalisation rouge Harmony style 6
10	XB6AV3BB	Unité de signalisation verte Harmony style 6

La direction et la vitesse du moteur sont fonction du cycle d'activité PWM donné par le signal de sortie du levier :

Condition	Résultat
Si le cycle d'activité > 52 %	Le moteur tourne **dans le sens des aiguilles d'une montre**, vitesse = (cycle d'activité - 50)*30 tr/min, l'unité de signalisation **verte** s'allume.
Si 48 % ≤cycle d'activité≤ 52 %	Le moteur **s'arrête**, l'unité de signalisation **rouge** s'allume.
Si le cycle d'activité < 48 %	Le moteur tourne **dans le sens inverse des aiguilles d'une montre**, vitesse = (50 - cycle d'activité)*30 tr/min, l'unité de signalisation **jaune** s'allume.

Un réseau CANopen est utilisé entre l'automate et le lecteur Altivar. Twido Extreme dispose d'un bus de terrain CANopen intégré. Ainsi, aucun module CANopen supplémentaire n'est requis.

Le lecteur Altivar est contrôlé par la macro DRIVE utilisée dans le programme de l'automate.

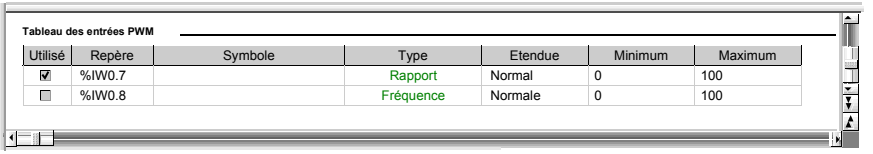
Création d'un exemple d'application

Suivez les étapes ci-dessous pour créer l'exemple précédemment décrit dans cette section à l'aide de TwidoSuite.

Etape	Action
1	Déclarez Twido Extreme dans la fenêtre Description à l'aide d'un glisser-déposer à partir du catalogue. Créez un réseau CANopen avec un lecteur ATV31 pour contrôler la vitesse et la direction du moteur. **Remarque** : La section Mise en œuvre du bus CANopen (voir *p. 266*) décrit la création de réseaux CANopen. En ce qui concerne Twido Extreme, un module maître CANopen n'est pas requis étant donné qu'il est intégré à Twido Extreme. D'autres fonctions CANopen pour Twido Extreme sont décrites à la section *p. 252*.
2	Configurez l'entrée PWM à l'aide de TwidoSuite pour convertir le signal PWM généré par un levier à axe unique en rapport.
3	Ecrivez un programme pour activer les sorties Twido Extreme en fonction de la valeur du rapport calculée à partir du signal d'entrée.
4	Connectez physiquement les composants matériels précédemment cités à l'automate Twido Extreme.
5	Transférez le programme du PC vers l'automate.

Configuration d'une entrée PWM

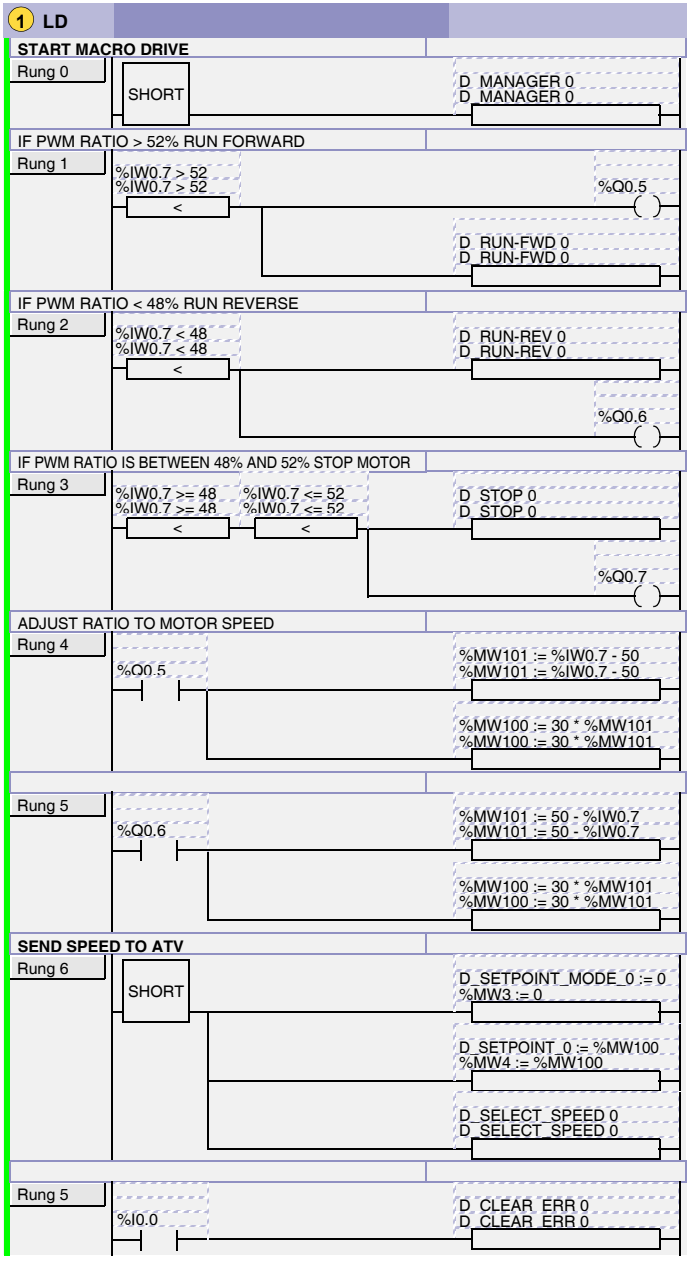
Dans **Programme** → **Configurer** → **Configurer le matériel** → **onglet Entrée** → **Tableau des entrées PWM**, sélectionnez **Rapport** dans le champ Type pour l'entrée %IW0.7, comme indiqué ci-dessous :



Note : Vous trouverez de plus amples détails sur ces champs de configuration à la section Configuration d'entrée PWM (voir *p. 182*).

Exemple de programmation d'une entrée PWM

Le programme est d'abord présenté en langage schéma à contact, puis sous forme de liste d'instructions. Le programme utilise la macro DRIVE pour contrôler le lecteur Altivar.



Programme en langage liste d'instructions, exemple d'application d'une entrée PWM :

```
----(* START MACRO DRIVE *)
0 LD 1
1 [ D_MANAGER 0 ]
----(* IF PWM RATIO > 52% RUN FORWARD *)
2 LD [ %IW0.0.7 > 52 ]
3 ST %Q0.0.5
4 [ D_RUN_FWD 0 ]
----(* IF PWM RATIO < 48% RUN REVERSE*)
5 LD [ %IW0.0.7 < 48 ]
6 [ D_RUN_REV 0 ]
7 ST %Q0.0.6
----(* IF PWM RATIO IS BETWEEN 48% AND 52 % STOP MOTOR *)
8 LD [ %IW0.0.7 >= 48 ]
9 AND [ %IW0.0.7 <= 52 ]
10 [ D_STOP 0 ]
11 ST %Q0.0.7
----(* ADJUST RATIO TO MOTOR SPEED *)
12 LD %Q0.0.5
13 [ %MW101 := %IW0.0.7 - 50 ]
14 [ %MW100 := 30 * %MW101 ]
----
15 LD %Q0.0.6
16 [ %MW101 := 50 - %IW0.0.7 ]
17 [ %MW100 := 30 * %MW101 ]
----(* SEND SPEED TO ATV *)
18 LD 1
19 [ %MW3 := 0 ]
20 [ %MW4 := %MW100 ]
21 [ D_SELECT_SPEED 0 ]
----(* RESET ERROR BIT *)
22 LD %I0.0.0
23 [ D_CLEAR_ERR 0 ]
```

Syntaxe d'E/S en langage schéma à contacts/liste d'instructions

Note : Dans les programmes en langage schéma à contacts, la syntaxe du repère d'E/S est %IW0.i (i=0...8) et %Q0.j (j=0...18), telle qu'elle apparaît dans les tableaux de configuration. Cependant, dans les programmes à liste d'instructions, la syntaxe %IW0.0.i et %Q0.0.j est utilisée pour faire référence à ces mêmes repères.

Connexions entrée/sortie

Connectez les entrées et les sorties du matériel :

- Connectez la sortie du levier à l'entrée Twido Extreme %IW0.7.
- Connectez l'entrée du lecteur Altivar via la boîte de raccordement au port CANopen Twido Extreme.
- Connectez les sorties %Q0.5, %Q0.6 et %Q0.7 Twido Extreme aux unités de signalisation pour activer la surveillance du système.
- Connectez le lecteur Altivar au moteur.
- Connectez le verrou électronique BlueTooth à Twido Extreme pour le transfert du programme.

Note : Pour plus d'informations à propos du levier à axe unique et des connexions E/S, reportez-vous à la rubrique Wiring Rules and Recommendations, Twido Extreme Hardware Guide.

8.3 Configuration des sorties de Twido Extreme

Présentation

Objet de cette section

Cette section décrit la configuration des sorties pour l'automate Twido Extreme.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration des sorties TOR de Twido Extreme	191
Configuration de la sortie du générateur d'impulsions (PLS) Twido Extreme	194
Configuration de la sortie PWM en mode Standard de Twido Extreme	200
Configuration de la sortie PWM en mode Hydraulique de Twido Extreme	209
Exemple de configuration de la sortie PWM hydraulique de Twido Extreme	216

Configuration des sorties TOR de Twido Extreme

Introduction

Cette section décrit la configuration des sorties TOR.

Sortie TOR

Une sortie TOR est le seul type de sortie de Twido Extreme.

Les sorties TOR peuvent être des sorties TOR standard ou liées à des blocs fonction (telles que PWM ou PLS) ou donner des informations sur l'état de l'automate (ce type de sortie est limité à 50 mA). Une sortie fonctionne en logique inversée : une valeur de 1 correspond à une tension de 0 (ou faible) et une valeur de 0 correspond à une tension élevée. Cela peut être utile pour certaines applications, comme les relais.

Repères de sortie TOR

Le tableau suivant présente les repères affectés aux sorties TOR Twido Extreme :

SORTIES TOR (19 au total)		
Type	Nombre max	Plage de repères
PLS/PWM	3	%Q0.0 - %Q0.2
Normal (1A)/Etat de l'automate (limité à 50 mA)	1	%Q0.3
Normal (1A)	1	%Q0.4
Normal (300 mA)	13	%Q0.5 - %Q0.17
Inversé	1	%Q0.18
Remarque : %Q0.3 est le seul repère pouvant être utilisé pour fournir l'état de l'automate. Ce repère peut également être utilisé en tant que repère TOR normal.		

Configuration de sortie TOR

Les sorties TOR utilisées dans ce programme en langage schéma à contact/liste d'instructions peuvent être affichées et configurées dans le volet Configuration du module de TwidoSuite (**Programme** → **Configurer** → **Configurer le matériel**).

Faites défiler l'onglet Sortie du volet Configuration du module qui répertorie les sorties TOR utilisées et disponibles, comme illustré ci-dessous :

Note : En mode connecté, les valeurs de sortie sont également affichées.

Les trois premières sorties %Q0.0 - %Q0.2 sont dédiées aux blocs fonction PLS/PWM.

Les sorties %Q0.3 - %Q0.17 sont des sorties TOR normales ayant différents niveaux de courant et de protection possibles.

La sortie %Q0.3 est utilisée pour donner l'état de l'automate.

La sortie %Q0.18 correspond à la sortie logique inversée. Une valeur de 1 correspond à une tension de 0 (ou faible) et une valeur de 0 correspond à une tension élevée).

Champs de configuration de sortie

Les champs de configuration de sortie affichés dans l'illustration ci-dessus sont détaillés dans le tableau suivant. Les champs grisés de ce tableau sont uniquement destinés à la consultation et ne peuvent pas être modifiés.

Champ	Valeurs possibles	Fonction
Utilisé	Case à cocher activée ou non.	Uniquement pour l'affichage. Affiche les sorties utilisées par le programme.
Repère	Repères de sortie %Q0.0-%Q0.18	Repère de chaque sortie TOR
Symbole	Valeur définie par l'utilisateur : 32 caractères alphanumériques maximum	Permet de donner un nom à la sortie. Ce champ peut être modifié. Une fois appliqué, ce nom est alors affiché dans le programme en langage schéma à contacts ou liste d'instructions.
Etat	Case à cocher : activée ou non Appliqué en sélectionnant et en cliquant sur Appliquer (ou en quittant la fenêtre, auquel cas il vous sera demandé de confirmer ou non l'application des modifications).	Utilisé pour indiquer l'état de l'automate. Si l'automate est en mode RUN, la sortie est définie sur 1. Si l'automate est en mode STOP ou en erreur, la sortie est définie sur 0.
Utilisé par	logique utilisateur	Uniquement pour l'affichage. Répertorie l'ensemble des blocs fonction ou logiques de programme utilisant cette sortie.

Configuration de la sortie du générateur d'impulsions (PLS) Twido Extreme

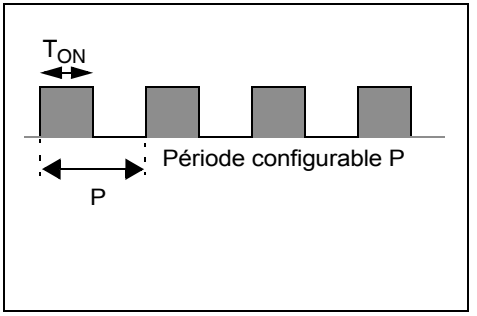
Introduction

Cette section décrit la configuration de la sortie PLS de l'automate Twido Extreme.

Sortie du générateur d'impulsions (PLS)

PLS est une fonction spéciale ayant %Q0.0 - %Q0.2 comme sorties dédiées sur un automate Twido Extreme. Un bloc fonction défini par l'utilisateur génère un signal d'impulsion sur ces sorties. Ce signal carré a une période P constante (configurable par l'utilisateur) associée à un cycle d'activité constant (non configurable). Le cycle d'activité du générateur PLS est défini en usine à 50%(T_{ON} / P).

Illustration du cycle d'activité PLS = 50%(T_{ON} / P) :



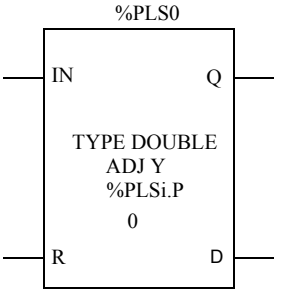
Twido Extreme prend en charge 3 générateurs PLS.

Sorties PLS/ PWM dédiées

Les sorties %Q0.0 - %Q0.2 sont dédiées aux blocs fonction %PLS ou %PWM. Par exemple, créer le bloc fonction %PLS*i* (*i*=0...2) affecte automatiquement la sortie %Q0.*i* à ce bloc fonction. Une fois que cette sortie a été utilisée pour %PLS*i*, elle ne peut plus être réutilisée par un bloc fonction %PWM*i* ou ailleurs dans le programme.

Bloc fonction %PLS

La figure ci-dessous présente la représentation d'un schéma à contacts d'un bloc fonction %PLS pour l'automate Twido Extreme.



Un bloc fonction %PLS dispose de plusieurs variables résumées dans le tableau suivant. La section Configuration d'un bloc fonction %PLS (voir *p. 197*) explique comment configurer ces variables.

Le bloc fonction %PLS*i* (*i* = 0...2) dispose des variables suivantes :

Objet	Description	Valeurs possibles	Accès en écriture
IN	Fonction Valider	0,1 Si IN=1, la génération d'impulsion est produite sur PLS*i*.Q SI IN=0, PLS*i*.Q est réglée sur 0	N
PLS*i*.R	Remis à 0	0,1 Si PLS*i*.R=1, les sorties %PLS*i*.Q et %PLS*i*.D sont réglées sur 0.	N
PLS*i*.Q	Génération dans le programme	0,1 Si PLS*i*.Q=1, le signal d'impulsion est généré au niveau de la sortie %Q0.*i* dédiée configurée.	N
PLS*i*.D	Fin de cycle	0,1 Si PLS*i*.D=1, la génération du signal est terminée. Le nombre voulu d'impulsions a été généré.	N
TYPE	Définit la plage possible pour le nombre d'impulsions pouvant être définis.	Simple ou double (mot)	Y
ADJUSTABLE	Détermine si la valeur de la période préréglée peut être modifiée.	Y/N Y permet la modification de la valeur préréglée.	Y

Remarque

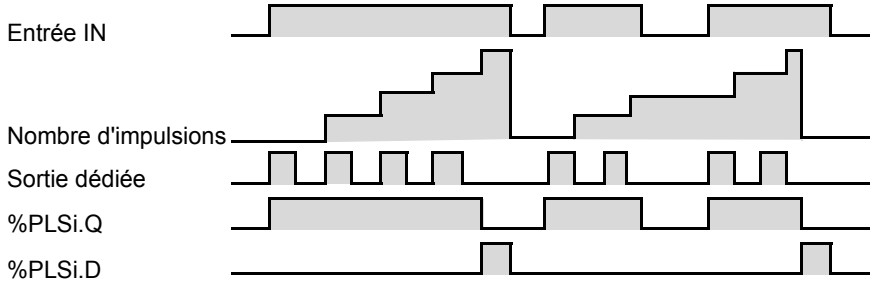
[1] La variable PLS*i*.N (ou ND pour un mot double) définit le nombre total d'impulsions souhaitées. Elle doit être définie dans le programme en langage schéma à contacts/liste d'instructions de l'utilisateur, un bloc opération par exemple, tel qu'indiqué ci-dessous à l'étape 1.

Objet	Description	Valeurs possibles	Accès en écriture
PLS0.P et PLS1.P	Période **préréglée** (calculée à partir de la fréquence définie par l'utilisateur dans le tableau de configuration.)	Plage de fréquences : 10...1 000 Hz ayant le résultat suivant : Plage de périodes : 100...10 000 (en 10 µs).	Y
PLS2.P	Période **préréglée** (calculée à partir de la fréquence définie par l'utilisateur dans le tableau de configuration.)	Plage de fréquences : 10...5 000 Hz ayant le résultat suivant : Plage de périodes : 20...10 000 (en 10 µs).	Y
PLS*i*.N[1]	Nombre total **d'impulsions** à générer.	0 ≤ PLS*i*.N ≤ 32 767	Y
PLS*i*.ND[1]	Nombre total **d'impulsions** (format mot double) à générer.	0 ≤ PLS*i*.ND ≤ 4 294 967 295	Y

Remarque

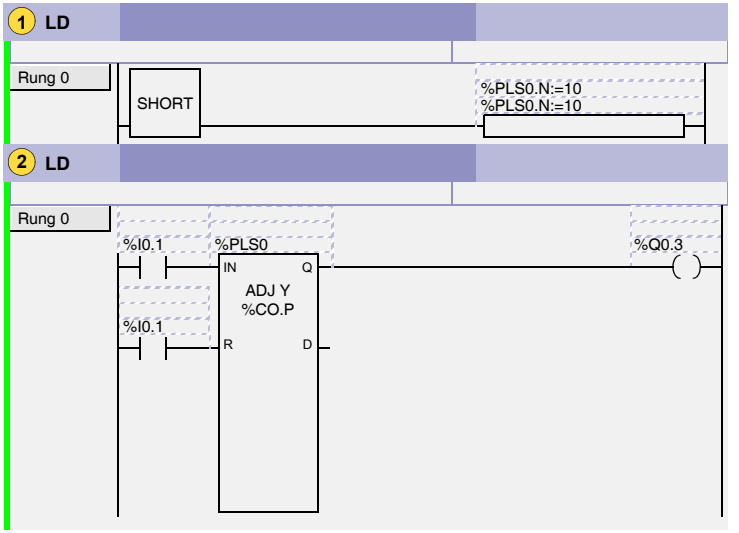
[1] La variable PLS*i*.N (ou ND pour un mot double) définit le nombre total d'impulsions souhaitées. Elle doit être définie dans le programme en langage schéma à contacts/liste d'instructions de l'utilisateur, un bloc opération par exemple, tel qu'indiqué ci-dessous à l'étape 1.

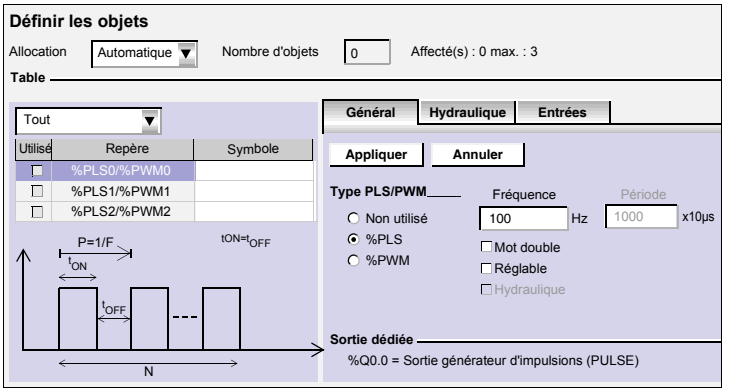
L'illustration suivante représente le diagramme d'impulsion d'un bloc fonction PLS.



Configuration d'un bloc fonction %PLS

Le tableau suivant explique comment configurer un bloc fonction %PLS.

Etape	Action
1	Créez un bloc fonction %PLS*i* dans l'éditeur ladder/list (*i* = 0..2) Blocs des schémas à contacts (voir *p. 390*).
2	**Exemple** : Le programme en langage schéma à contacts avec bloc fonction dans la section 2 et une variable %PLSi.N configurée à la section 1 (1).  (1) La variable PLS*i*.N (ou ND pour un mot double) définit le nombre total d'impulsions souhaitées. Elle doit être définie dans le programme en langage schéma à contacts/liste d'instructions, un bloc opération par exemple, tel qu'indiqué à l'étape 1. La valeur par défaut est réglée sur 0. Pour produire un nombre illimité d'impulsions, réglez %PLS*i*.N ou %PLS*i*.ND sur zéro.

Etape	Action
3	**Exemple** : Même exemple ci-dessous avec un programme liste d'instructions (1). **1 IL** 0 LD 1 1 [%PLS0.N := 10] **2 IL** 0 BLK %PLS0 1 LD %I0.0.1 2 IN 3 LD %I0.0.1 4 R 5 OUT_BLK 6 LD Q 7 ST %Q0.0.3 8 END_BLK (1) La variable PLS*i*.N (ou ND pour un mot double) définit le nombre total d'impulsions souhaitées. Elle doit être définie dans le programme en langage schéma à contacts/liste d'instructions, un bloc opération par exemple, tel qu'indiqué à l'étape 1. La valeur par défaut est réglée sur 0. Pour produire un nombre illimité d'impulsions, réglez %PLS*i*.N ou %PLS*i*.ND sur zéro.
4	Ouvrez la table de configuration PLS (voir *p. 199*). **Résultat :**  Le repère %PLS*i* qui vous avez précédemment défini pour le bloc fonction dans l'éditeur ladder s'affiche ici (avec la case Utilisée cochée). Vous pouvez définir jusqu'à 3 blocs fonction %PLS.
5	Cliquez sur la rangée dans le volet gauche correspondant au %PLS*i* que vous souhaitez configurer. Si la sortie dédiée Q0.*i* a déjà été utilisée ailleurs dans le programme, un message d'erreur s'affiche vous indiquant que la configuration de ce %PLS est impossible. Dans ce cas, reprenez votre programme et affectez un autre %PLS ou %Q.

Etape	Action
6	Seul l'onglet Général s'applique aux blocs fonction %PLS. Les autres onglets (Hydraulique et Entrées) sont uniquement utilisés pour les blocs fonction %PWM et ne sont pas disponibles pour les %PLS. Dans le volet droit (onglet Général) : • Sélectionnez le type **%PLS**. • Entrez une **Fréquence**(2) (3). Pour %PLS0 et %PLS1 : Fréquence 10 ... 1 000 Hz => Période 100 ... 10 000 Pour %PLS2 : Fréquence 10 ... 5 000 Hz => Période 20 ... 10 000 • Cochez ou décochez la case **Mot double**. • Cochez ou décochez la case **Réglable**. Le **nombre d'impulsions** ne peut pas être configuré dans cette fenêtre.(1) (1) La variable PLS*i*.N (ou ND pour un mot double) définit le nombre total d'impulsions souhaitées. Elle doit être définie dans le programme en langage schéma à contacts/liste d'instructions, un bloc opération par exemple, tel qu'indiqué à l'étape 1. (2) Saisir une valeur de fréquence hors de la plage entraîne un message d'erreur. (3) La période P (exprimée en 10 µs) est calculée à partir de la valeur de la fréquence F que vous avez entrée (P=1/F). Une fréquence de 10 engendre une période de 10 000 ; une fréquence de 1 000 engendre une période de 100.
7	Pour configurer toutes les autres sorties PLS requises (y compris celles qui ne sont pas encore utilisées dans le programme), répétez les étapes 3 et 4. Lors de la sélection d'un autre %PLS, vous devez confirmer si vous souhaitez ou non appliquer les modifications. Cliquez sur "Oui".
8	Cliquez sur **Appliquer** (ou quittez la fenêtre. Confirmez si vous souhaitez ou non appliquer ces modifications).
9	Si vous revenez à l'écran précédent à l'aide du bouton , les fenêtres précédentes s'affichent dans leur état premier.

Ouverture de la table de configuration %PLS

Le tableau ci-dessous indique comment ouvrir la table de configuration %PLS.

Etape	Action
1	Pour ouvrir la table de configuration %PLS à partir de la fenêtre **Programme** : Cliquez deux fois sur le bloc fonction %PLS dans l'Editeur Ladder.
2	Pour ouvrir une table de configuration %PLS à partir d'un autre emplacement de TwidoSuite : **1.** Sélectionnez **Programme** → **Configurer** → **Configurer les données** **2.** Sélectionnez **Objets d'E/S** dans Catégories d'objets. **3.** Sélectionnez **%PLS/%PWM** dans Objets d'E/S.

Configuration de la sortie PWM en mode Standard de Twido Extreme

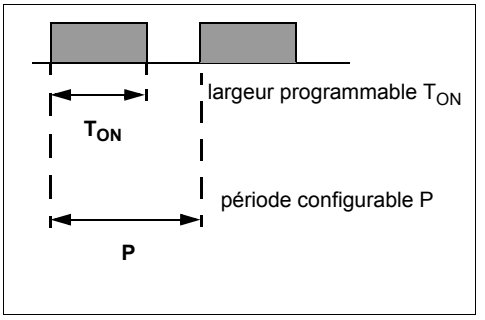
Introduction

Cette section explique comment configurer les sorties PWM en mode Standard.

Sortie PWM

PWM est une fonction spéciale qui peut être affectée à une sortie. Ce signal rectangulaire a une période P constante (configurable par l'utilisateur) avec la possibilité de faire varier la largeur de l'impulsion T_{ON} et ainsi le cycle d'activité (T_{ON} / P).

Illustration du cycle d'activité PWM :



Repères de sortie PWM

Vous pouvez configurer jusqu'à 3 sorties PWM.

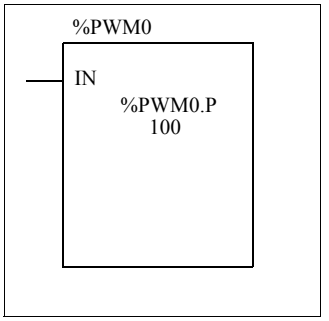
PWM est affectée à la sortie %Q0.0 à %Q0.2 :

Repère PWM	Sortie dédiée
%PWM0	%Q0.0
%PWM1	%Q0.1
%PWM2	%Q0.2

Les sorties %Q0.0 - %Q0.2 sont dédiées au bloc fonction %PWM ou %PLS. Par exemple, créer le bloc fonction %PWM*i* (i=0...2) affecte automatiquement la sortie %Q0.*i* à ce bloc fonction. Une fois que cette sortie a été utilisée pour %PWMi, elle ne peut plus être réutilisée par un bloc fonction %PLS*i* ou ailleurs dans le programme.

Bloc fonction %PWM

La figure ci-dessous montre la représentation d'un schéma à contacts d'un bloc fonction %PWM pour l'automate Twido Extreme.



Un bloc fonction %PWM dispose de plusieurs variables résumées dans le tableau suivant. La section Configuration d'un bloc fonction %PWM (voir *p. 204*) décrit comment configurer ces variables.

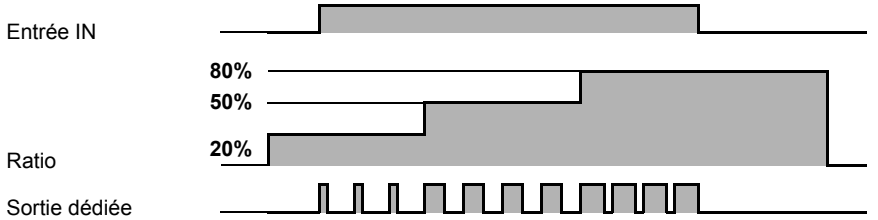
Le bloc fonction %PWM*i* (*i* = 0...2) dispose des variables suivantes :

Objet	Description	Valeurs possibles	Accès en écriture
IN	Fonction Valider	0,1 Si IN=1, la génération des impulsions est produite au niveau de la sortie dédiée %Q0 configurée.*i* (*i*=0,1,2). SI IN=0, la voie de sortie est paramétrée sur 0.	N
%PWMi.R	Cycle d'activité Le fait de modifier le cycle d'activité de PWMi.R dans le programme module la largeur du signal.	Cette valeur donne le pourcentage du signal à l'état 1 au cours d'une période P. La largeur d'impulsion T_{ON} équivaut ainsi à : T_{ON} = P * (%PWMi.R/100). (P étant la période de 10 µs). La valeur par défaut est 0 et les valeurs supérieures à 100 sont considérées comme étant égales à 100. %PWMi.R doit être défini dans le programme en langage schéma à contacts/liste d'instructions via un bloc opération par exemple, tel qu'indiqué ci-dessous à l'étape 1.	Y
PWM0.P et PWM1.P	Période **préréglée** (calculée à partir de la fréquence définie par l'utilisateur dans le tableau de configuration.)	Plage de fréquences : 10...1 000 Hz ayant le résultat suivant : Plage de périodes : 100...10 000 (en 10 µs).	Y

Objet	Description	Valeurs possibles	Accès en écriture
PWM2.P	Période **préréglée** (calculée à partir de la fréquence définie par l'utilisateur dans le tableau de configuration.)	Plage de fréquences : 10...5 000 Hz ayant le résultat suivant : Plage de périodes : 20...10 000 (en 10 µs).	Y
ADJUSTABLE	Détermine si la valeur de la période préréglée peut être modifiée.	Y/N Y permet la modification de la valeur préréglée.	Y

Bloc fonction PWM avec des cycles d'activité qui varient

L'illustration suivante représente le diagramme d'impulsion du bloc fonction PWM avec des cycles d'activité qui varient.



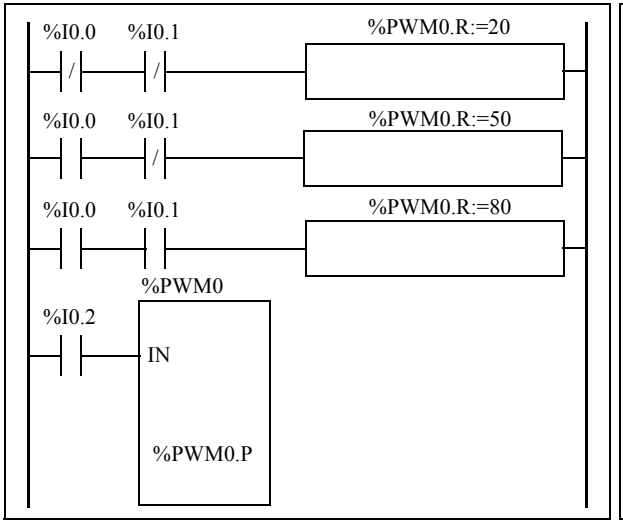
Dans cet exemple de programmation, la largeur du signal est modifiée par le programme en fonction de l'état des entrées %I0.0 et %I0.1 de l'automate.

Si %I0.1 et %I0.2 sont réglés sur 0 et le rapport %PWM0.R sur 20 %, la durée du signal à l'état 1 est alors égale à : 20 % x 500 ms = 100 ms.

Si %I0.0 est réglé sur 0, %I0.1 sur 1 et le rapport %PWM0.R sur 50 %, la durée est alors de 250 ms.

Si %I0.0 et %I0.1 sont réglés sur 1 et le rapport %PWM0.R sur 80 % (durée de 400 ms).

Exemple de programmation :

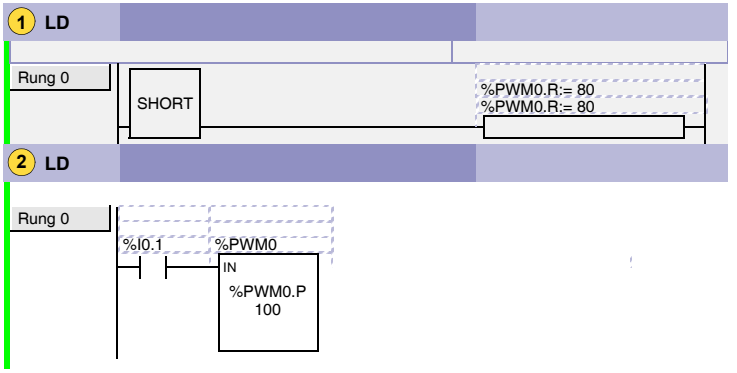



```
LDN     %I0.0.0
ANDN    %I0.0.1
[%PWM0.R:=20]
LD      %I0.0.0
ANDN    %I0.0.1
[%PWM0.R:=50]
LD      %I0.0.0
AND     %I0.0.1
[%PWM0.R:=80]
BLK     %PWM0
LD      %I0.0.2
IN
END_BLK
```

Configuration d'un bloc fonction %PWM

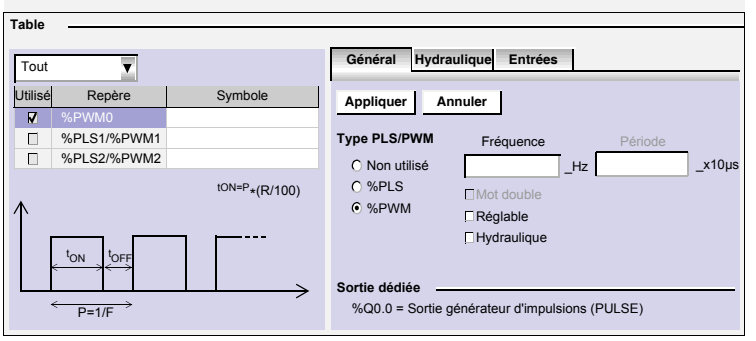
Le tableau suivant explique comment configurer un bloc fonction %PWM.

Etape	Action
1	Créez un bloc fonction %PWM*i* dans l'éditeur ladder/list (*i* = 0..2) Blocs des schémas à contacts (voir *p. 462*). **Exemple** : Le programme en langage schéma à contacts avec bloc fonction dans la section 2 et une variable %PWM0.R de cycle d'activité configurée à la section 1.



Même exemple ci-dessous avec un programme liste d'instructions :

```
1 IL
0       LD          1
1       [ %PWM0.R:= 80 ]

2 IL
0       BLK         %PWM0
1       LD          %I0.0.1
2       IN
3       END_BLK
```

Etape	Action
2	Ouvrez la table de configuration PMW (voir *p. 206*). **Résultat :**  Le repère %PWM*i* que vous avez précédemment défini pour le bloc fonction dans l'éditeur ladder s'affiche ici (avec la case Utilisée cochée). Vous pouvez définir jusqu'à 3 blocs fonction %PWM. La section Configuration de la sortie PMW (voir *p. 200*) décrit ces champs.
3	Cliquez sur la rangée dans le volet gauche correspondant au %PLS*i* que vous souhaitez configurer. Si la sortie dédiée Q0.*i* a déjà été utilisée ailleurs dans le programme, un message d'erreur s'affiche vous indiquant que la configuration de ce %PWM est impossible. Dans ce cas, reprenez votre programme et affectez un autre %PWM ou %Q.
4	L'onglet Général avec l'option PWM sélectionnée correspond à la sortie %PWM. L'onglet Hydraulique correspond à la sortie %PWM en mode hydraulique (voir *p. 209*). Dans le volet droit (onglet Général) : ● Sélectionnez le type **%PWM**. ● Entrez une **Fréquence**(1) (2) : Pour %PLS0 et %PLS1 : Fréquence 10 ... 1 000 Hz => Période 100 ... 10 000 Pour %PLS2 : Fréquence 10 ... 5 000 Hz => Période 20 ... 10 000 ● Cochez ou décochez la case **Réglable**. Le **cycle d'activité** ne peut pas être configuré dans cette fenêtre. Il doit être défini dans le programme en langage schéma à contacts/liste d'instructions via un bloc opération par exemple, tel qu'indiqué à l'étape 1. (1) La saisie d'une valeur de fréquence hors-plage entraîne un message d'erreur. (2) La période P (exprimée en 10 µs) est calculée à partir de la valeur de la fréquence F que vous avez entrée (P=1/F). Une fréquence de 10 engendre une période de 10 000 ; une fréquence de 1 000 engendre une période de 100.
5	Pour configurer toutes les autres sorties PWM requises (y compris celles qui ne sont pas encore utilisées dans le programme), répétez les étapes 3 et 4. Lors de la sélection d'un autre %PWM, vous devez confirmer si vous souhaitez ou non appliquer les modifications. Cliquez sur "Oui".
6	Cliquez sur **Appliquer** (ou quittez la fenêtre, vous êtes alors invité à confirmer l'application de ces modifications).
7	Si vous revenez à l'écran précédent à l'aide du bouton , les fenêtres précédentes s'affichent dans leur état antérieur.

Ouverture de la table de configuration %PWM

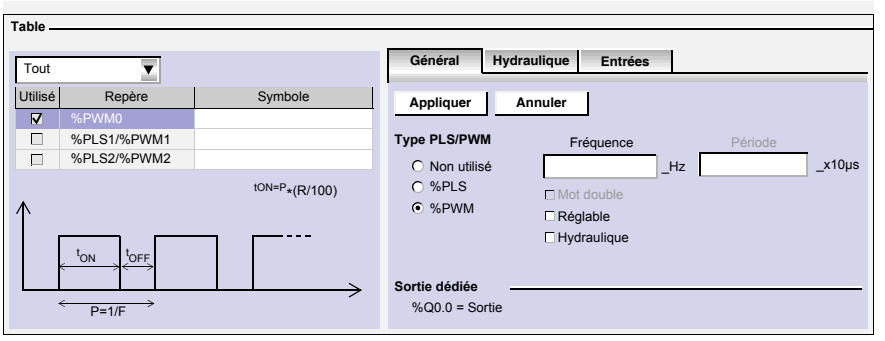
Le tableau ci-dessous indique comment ouvrir une table de configuration %PWM.

Etape	Action
1	Pour ouvrir une table de configuration %PWM à partir de la fenêtre **Programme** : Cliquez deux fois sur le bloc fonction %PWM dans l'Editeur Ladder.
2	Pour ouvrir une table de configuration %PWM à partir d'un autre emplacement de TwidoSuite : **1.** Sélectionnez **Programme** → **Configurer** → **Configurer les données** **2.** Sélectionnez **Objets d'E/S** dans Catégories d'objets. **3.** Sélectionnez **%PLS/%PWM** dans Objets d'E/S.

Configuration de sortie PWM

Les sorties PWM peuvent être affichées et configurées dans **Programme** → **Configurer** → **Configurer les données** → **Objets d'E/S** → **%PLS/%PWM**.

L'onglet Général de la **Table de configuration PWM** est présenté ci-dessous :



Champs de configuration de l'onglet Général de la sortie PWM

Les champs de configuration de la sortie PWM sont décrits dans le tableau suivant. Comme indiqué, certains champs de ce tableau sont uniquement destinés à l'affichage (lecture seule) et ne peuvent pas être modifiés.

Champ	Valeurs possibles	Fonction
Zone Filtrage	Liste déroulante proposant les options suivantes : ● Tout ● Utilisé ● Inutilisé	Vous permet d'afficher/masquer les objets utilisés/inutilisés.
Utilisé	Case à cocher activée ou non. Une case cochée correspond à un élément utilisé. Une case non cochée correspond à un élément inutilisé.	Uniquement pour l'affichage. Affiche les sorties PWM utilisées par le programme. Ce champ ne peut pas être modifié.
Repère	Repères de sortie PWM : ● %PWM0 ● %PWM1 ● %PWM2	Repère de chaque sortie PWM.
Symbole	Valeur définie par l'utilisateur : 32 caractères alphanumériques maximum	Permet de donner un nom à la sortie PWM. Ce champ peut être modifié. Une fois appliqué, ce nom est alors affiché dans le programme en langage schéma à contacts ou liste d'instructions.
Type PLS/ PWM	3 options au choix : ● Non utilisé ● %PLS ● %PWM	Permet de choisir une génération PLS ou PWM.
Fréquence	Valeur choisie par l'utilisateur parmi la plage suivante : ● Pour %PLS0 et %PLS1 : 10 ... 1 000 Hz en mode standard. Pour %PLS2 : 10 ... 5 000 Hz en mode standard. ● 50 Hz ... 400 Hz en mode hydraulique (voir *p. 209*).	Fréquence du signal de sortie PWM.
Période	Dérivée de la fréquence (P=1/F). **Remarque :** la période peut être modifiée dans le programme de l'application en utilisant le paramètre %PLS*i*.P.	Période (en 10 µs) du signal de sortie PWM.

Champ	Valeurs possibles	Fonction
Réglable	Case à cocher activée ou non. Si elle est cochée, il est possible de modifier la période via le programme TwidoSuite ou l'éditeur de tables d'animation en utilisant le paramètre %PLS*i*.P Si elle n'est pas cochée, il est impossible de modifier la période en utilisant le paramètre %PLS*i*.P	Définit si la valeur de la période préréglée peut être modifiée.
Hydraulique	Case à cocher (reportez-vous à la section *p. 209*) **Remarque :** cocher la case Hydraulique active les onglets **Hydraulique** et **Entrée**.	Activer/désactiver le mode Hydraulique.
Sortie dédiée	Sortie dédiée PWM : • %Q0.0 • %Q0.1 • %Q0.2	Sortie dédiée à PWM.
Appliquer/ Annuler	Cliquez sur : • **Appliquer** pour confirmer et enregistrer les modifications. • **Annuler** pour annuler les modifications.	Enregistrer ou annuler les modifications d'un projet TwidoSuite.

Cycle d'activité

Vous pouvez définir et modifier le cycle d'activité (R) du programme utilisateur en utilisant le paramètre %PWM*i*.R.

Configuration de la sortie PWM en mode Hydraulique de Twido Extreme

Introduction

Cette section décrit la configuration de la sortie PWM en mode Hydraulique.

Sortie PWM en mode Hydraulique

PWM est une fonction spéciale qui peut être affectée à une sortie. La période de ce signal rectangulaire est constante et configurable par l'utilisateur, avec la possibilité de varier le cycle d'activité. La section Sortie PWM (voir *p. 200*) propose plus de détails à ce sujet.

PWM en mode Hydraulique permet à Twido Extreme de contrôler le système hydraulique.

Repères de sortie PWM

Vous pouvez configurer jusqu'à 3 sorties PWM.

PWM est affectée à la sortie %Q0.0 à %Q0.2 :

Repère PWM	Sortie dédiée
%PWM0	%Q0.0
%PWM1	%Q0.1
%PWM2	%Q0.2

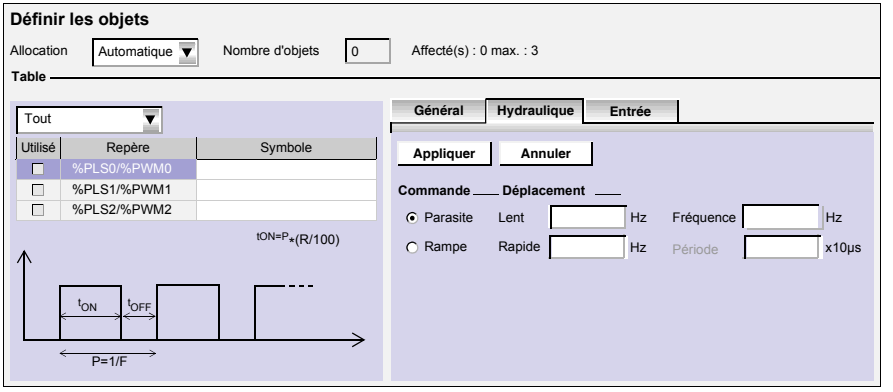
Les sorties %Q0.0.0 - %Q0.0.2 sont dédiées au bloc fonction %PWM ou %PLS. Par exemple, créer le bloc fonction %PWM*i* (i=0...2) affecte automatiquement la sortie %Q0.*i* à ce bloc fonction. Une fois que cette sortie a été utilisée pour %PWMi, elle ne peut plus être réutilisée par un bloc fonction %PLS*i* ou ailleurs dans le programme.

Note : En mode hydraulique, les autres entrées et sorties TOR peuvent également être utilisées. Cependant, les trois sorties %Q0.0-%Q0.2 sont réservées à la sortie du signal de %PWM0-%PWM2 et ne peuvent pas être réutilisées.

Configuration de l'onglet Hydraulique de la sortie PWM

Les sorties PWM peuvent être affichées et configurées dans **Programme** → **Configurer** → **Configurer les données** → **Objets d'E/S** → **%PLS/%PWM** → **onglet Hydraulique**.

L'onglet Hydraulique de la **Table de configuration PWM** est présenté ci-dessous :



Champs de configuration de l'onglet Hydraulique de la sortie PWM

Les champs de configuration de l'onglet Hydraulique de la sortie PWM sont décrits dans le tableau suivant.

Champ	Valeurs possibles	Fonction
Commande	Sélectionnez l'une des deux options suivantes : ● Dither ● Rampe	voir *p. 211* et *p. 211*
Déplacement	**Mode Rampe** : Définissez la valeur dans la plage [0 s...9 s] pour : ● rampe lente ● rampe rapide Mode Dither : Définissez la valeur dans la plage [2 Hz...400 Hz] pour : ● parasite lent ● parasite rapide	Définir la durée de la rampe montante et descendante (mode rampe) ou pour définir la fréquence de parasite (mode parasite).
Fréquence/ Période	Valeur définie par l'utilisation parmi les plages suivantes : ● [50 Hz...400 Hz en mode hydraulique]	Modifier la valeur de la fréquence définie dans l'onglet Général. La période est calculée à partir de la fréquence saisie par l'utilisateur et ne peut pas être modifiée directement.
Appliquer/ Annuler	Cliquez sur : ● **Appliquer** pour confirmer et enregistrer les modifications. ● **Annuler** pour ignorer les modifications.	Enregistrer ou annuler les modifications d'un projet TwidoSuite.

Définition de Dither

Le frottement et l'hystérèse peuvent rendre le contrôle d'une valve hydraulique inégal et imprévisible :

- le frottement peut éviter au tiroir de commande de bouger lors de petites modifications du signal d'entrée et
- l'hystérèse peut entraîner un décalage du tiroir pour la même entrée du signal de commande et ce, en fonction de l'augmentation ou de la diminution de la modification.

Il existe deux définitions de parasite :

- Dans la zone constante (voir *p. 214*), un petit mouvement rapide du tiroir à un endroit souhaité a pour but de le mettre en mouvement afin d'éviter le frottement. Mouvement limité à 5 % de la valeur de déplacement totale autour de l'emplacement souhaité.
- Pour une rampe montante ou descendante (voir *p. 214*), la position de la valve modifie la fréquence. Par exemple, pour un parasite d'une fréquence de 100 Hz, la valve modifie sa position toutes les 10 ms.

Note : L'amplitude du Dither doit être assez importante et la fréquence assez faible pour permettre au tiroir de répondre. Cependant, l'amplitude du parasite ne doit pas être trop importante et la fréquence ne doit pas être trop faible afin de ne pas provoquer d'impulsion au niveau de la sortie hydraulique.

Définition de rampe

Les rampes sont utilisées pour ralentir la réponse de la commande de la valve par rapport à la modification de la commande d'entrée. Ceci engendre une légère transition lors d'une modification rapide d'un signal d'entrée de commande.

Les rampes n'ont aucun effet si la modification du signal d'entrée est plus lente que le paramètre de la rampe.

Les rampes peuvent être fixes ou réglables. Pour les paramètres rapides (ou lents), la courbe de la rampe est la même en montée qu'en descente.

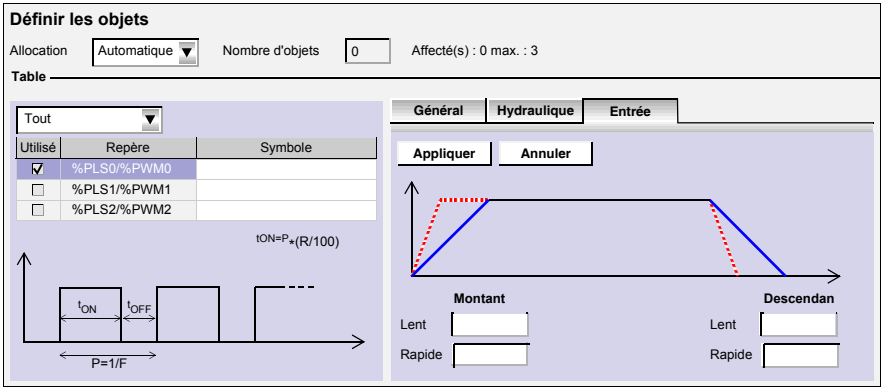
Twido Extreme vous permet de configurer un total de quatre rampes (2 montantes rapides/lentes et 2 descendantes rapides/lentes). Les rampes indépendantes doivent avoir des commandes de rampes séparées pour la montée et la descente (voir ci-dessous). Vous devez au moins définir une rampe montante et une rampe descendante.

Des rampes symétriques ont la même courbe montante et descendante. Un contrôle séparé de l'accélération et de la décélération peut être obtenu à l'aide de commandes de rampe distinctes pour chaque rampe. Si des commandes de valves bidirectionnelles de bobines sont utilisées avec quatre potentiomètres, vous pouvez contrôler 4 rampes (montante lente/rapide et descendante lente/rapide).

Configuration d'une rampe PWM

Les rampes PWM peuvent être affichées et configurées dans **Programme** → **Configurer** → **Configurer les données** → **Objets d'E/S** → **%PLS/%PWM** → **onglet Entrée**.

L'onglet Entrée de la **Table de configuration PWM** est présenté ci-dessous :



Champs de configuration de l'onglet Entrée de la rampe PWM

Les champs de configuration de la rampe PWM sont décrits dans le tableau suivant. L'onglet Entrée est uniquement disponible si la case Hydraulique est cochée dans l'onglet Général. Saisir des repères dans ces champs vous permet de définir les rampes.

Champ	Valeurs possibles	Fonction
Rampe montante lente	Sélectionnez le déclenchement de la rampe montante lente : ● Bit mémoire %M*i* (*i*=0...99) ● Entrée numérique %I0.*j* (*j*=0...19) ● Sortie numérique %Q0.*k* (*k*=0...18))	Lorsque le déclenchement est réglé sur 1, la transition de la rampe montante lente commence.
Rampe montante rapide	Sélectionnez le déclenchement de la rampe montante rapide : ● Bit mémoire %M*i* (*i*=0...99 ● Entrée numérique %I0.*j* (*j*=0...19) ● Sortie numérique %Q0.*k* (*k*=0...18))	Lorsque le déclenchement est réglé sur 1, la transition de la rampe montante rapide commence.
Rampe descendante lente	Sélectionnez le déclenchement de la rampe descendante lente : ● Bit mémoire %M*i* (*i*=0...99 ● Entrée numérique %I0.*j* (*j*=0...19) ● Sortie numérique %Q0.*k* (*k*=0...18))	Lorsque le déclenchement est réglé sur 1, la transition de la rampe descendante lente commence.
Rampe descendante rapide	Sélectionnez le déclenchement de la rampe descendante rapide : ● Bit mémoire %M*i* (*i*=0...99) ● Entrée numérique %I0.*j* (*j*=0...19) ● Sortie numérique %Q0.*k* (*k*=0...18)	Lorsque le déclenchement est réglé sur 1, la transition de la rampe descendante rapide commence.
Appliquer/ Annuler	Cliquez sur : ● **Appliquer** pour confirmer et enregistrer les modifications. ● **Annuler** pour ignorer les modifications.	Enregistrer ou annuler les modifications d'un projet TwidoSuite.

Priorité de la rampe

Si les deux déclenchements rapide et lent sont définis simultanément pour la même rampe, l'ordre de priorité défini suivant protège la valve :

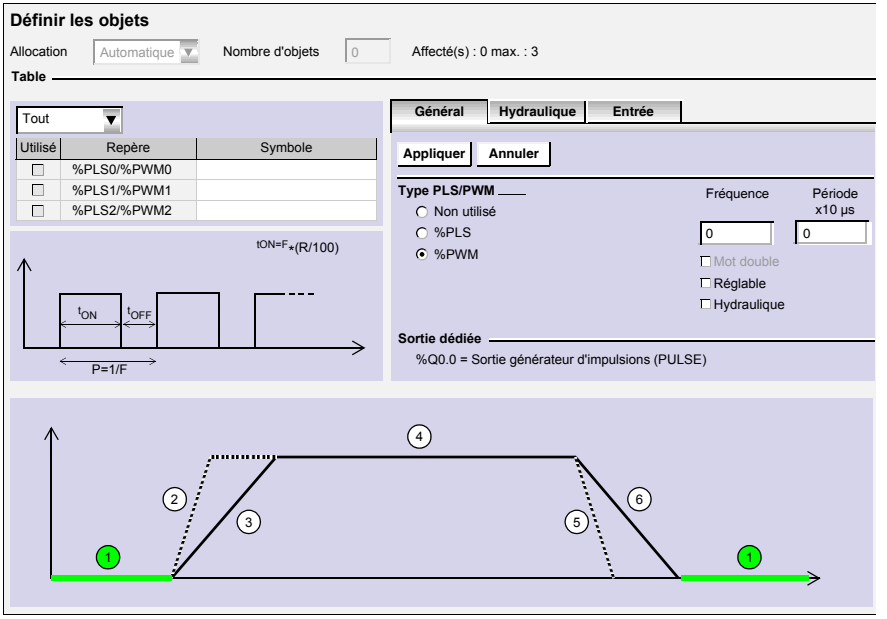
- Descente rapide
- Descente lente
- Montée lente
- Montée rapide

Cycle d'activité

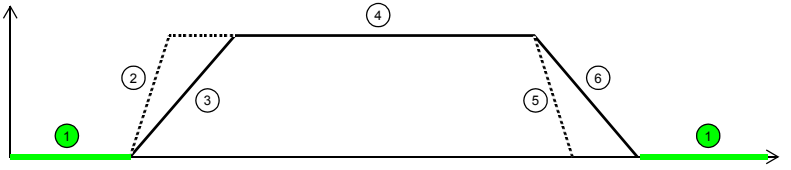
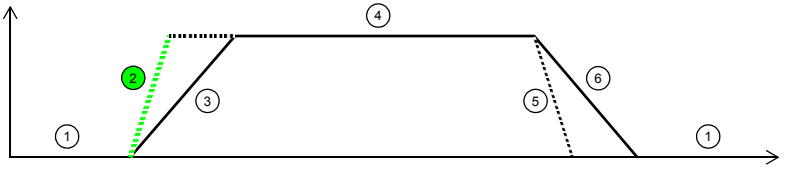
Vous pouvez définir et modifier le cycle d'activité (R) du programme utilisateur (en langage schéma à contacts ou liste d'instructions) en utilisant le paramètre %PWMi.R. Pour les applications hydrauliques, le cycle d'activité doit être compris dans la plage : $5\% \leq R \leq 95\ \%$.

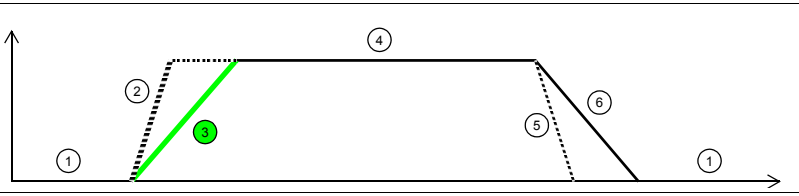
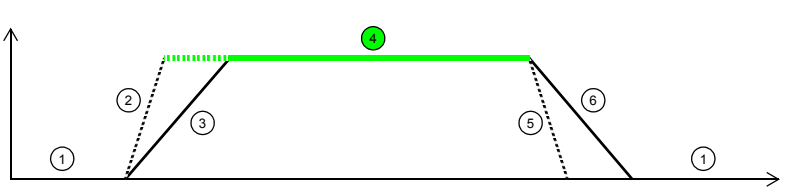
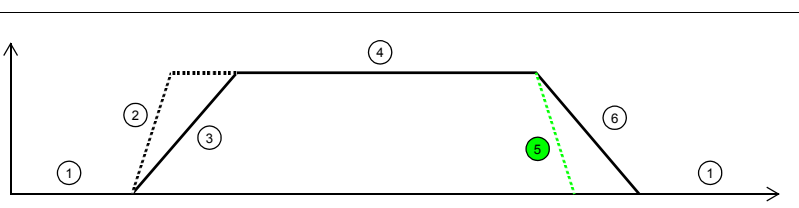
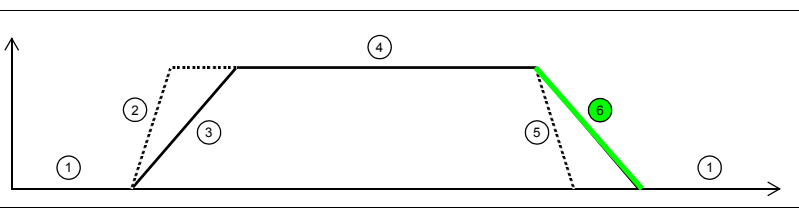
Sortie PWM hydraulique en mode connecté

En mode connecté, vous pouvez surveiller l'état de la rampe de sortie PWM dans **Programme → Mise au point → Surveiller la configuration logicielle**:



Il existe 6 états différents du signal PWM en mode connecté, comme indiqué ci-dessous :

Numéro de l'état	Etat	Schéma connecté
1	Angle mort	
2	Transition de la rampe montante rapide	

Numéro de l'état	Etat	Schéma connecté
3	Transition de la rampe montante lente	
4	Zone constante	
5	Transition de la rampe descendante rapide	
6	Transition de la rampe descendante lente	

Effet des Twido Extreme Modes de marche des sorties PWM

Etat des bits système	Mode de marche	Effet
%S0=1	Redémarrage à froid	La sortie est réinitialisée à son état initial et le signal de sortie est nul.
%S1=1	Redémarrage à chaud	La sortie redémarre au même état qu'elle avait avant la coupure secteur.
%S9=1	Automate en mode STOP ou sorties réinitialisées en mode RUN	Les sorties %Q0.0, %Q0.1 et %Q0.2 sont réglées sur 0 sans tenir compte de l'état du bit système %S8.

Exemple de configuration de la sortie PWM hydraulique de Twido Extreme

Vue d'ensemble

Cette section explique, à travers un exemple d'application, comment configurer une sortie PWM en mode hydraulique.

Processus de configuration d'une sortie PWM hydraulique

Suivez les étapes ci-dessous pour configurer la sortie PWM de manière à ce qu'elle contrôle un système hydraulique :

Etape	Description
1	Configurez l'onglet Général de la sortie PWM.
2	Configurez l'onglet Hydraulique de la sortie PWM.
3	Configurez l'onglet Entrée de la sortie PWM.

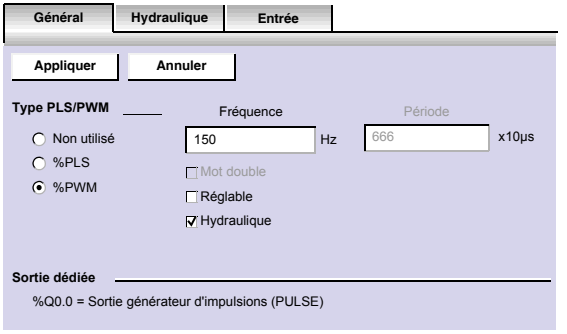
Accès à la fenêtre de configuration de la sortie PWM

Vous pouvez accéder à la fenêtre de configuration de la sortie soit :

- en cliquant deux fois sur le bloc fonction PWM d'un programme en langage schéma à contact,
- soit en sélectionnant **Programme** → **Configurer** → **Configurer les données** → **Objets d'E/S** → **%PLS/%PWM**.

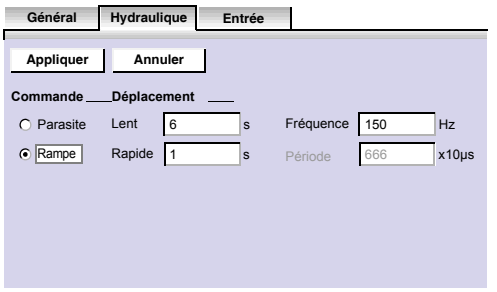
Configuration de l'onglet Général de la sortie PWM

Configurez l'onglet Général de la sortie PWM de la manière suivante :

Etape	Action
1	Sélectionnez %PWM dans le champ Type PLS/PWM.
2	Définissez la fréquence sur 150 Hz. **Résultat :** la période de x10 µs est calculée à partir de la fréquence (P=1/F).
3	Cochez la case Hydraulique pour permettre l'accès aux onglets Hydraulique et Entrée.
4	**Résultat :** 

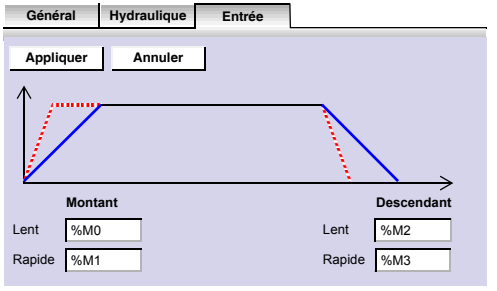
Configuration de l'onglet Hydraulique de la sortie PWM

Configurez l'onglet Hydraulique de la sortie PWM de la manière suivante :

Etape	Action
1	Sélectionnez Rampe dans le champ Commande.
2	Définissez le Déplacement lent sur 6 s.
3	Définissez le Déplacement rapide sur 1 s.
4	Si nécessaire, modifiez la fréquence précédemment définie.
5	**Résultat :** 

Configuration de l'onglet Entrée de la sortie PWM

Configurez l'onglet Entrée de la sortie PWM de la manière suivante :

Etape	Action
1	Définissez la rampe ascendante lente.
2	Définissez la rampe ascendante rapide.
3	Définissez la rampe descendante lente.
4	Définissez la rampe descendante rapide.
5	Cliquez sur Appliquer pour enregistrer les modifications apportées à tous les onglets.
6	**Résultat :** 

Mise en œuvre du bus AS-Interface V2

9

Présentation

Objet de ce chapitre

Ce chapitre fournit les informations sur la mise en œuvre logicielle du module maître AS-Interface TWDNOI10M3 et de ses esclaves.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation du bus AS-Interface V2	220
Description fonctionnelle générale	221
Principes de mise en œuvre logicielle	224
Description de l'écran de configuration du bus AS-Interface	225
Configuration du bus AS-Interface	227
Description de la fenêtre AS-Interface en mode connecté	233
Modification de l'adresse d'un esclave	236
Mise à jour de la configuration du bus AS-Interface en mode connecté	238
Adressage automatique d'un esclave AS-Interface V2	242
Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante	243
Remplacement automatique d'un esclave AS-Interface V2 défectueux	244
Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2	245
Programmation et diagnostic du bus AS-Interface V2	246
Mode de fonctionnement du module interface bus AS-Interface V2	250

Présentation du bus AS-Interface V2

Présentation

Le bus AS-Interface (Actuator Sensor-Interface) permet l'interconnexion, sur un câble unique, de capteurs/actionneurs au niveau le plus bas de l'automatisation.

Ces capteurs/actionneurs seront définis dans la documentation comme **périphériques esclaves**.

La mise en œuvre de AS-Interface nécessite de définir le contexte physique de l'application dans laquelle il sera intégré (bus d'expansion, alimentation, processeur, modules, périphériques esclaves AS-Interface connectés sur le bus) puis d'en assurer sa mise en œuvre logicielle.

Cette mise en œuvre logicielle sera réalisée depuis les différents éditeurs de TwidoSuite :

- soit en mode local,
- soit en mode connecté.

Bus AS-Interface V2

Le module maître AS-interface **TWDNOI10M3** intègre les fonctionnalités suivantes :

- Profil M3 : ce profil offre toutes les fonctionnalités définies par la norme AS-Interface V2, mais ne prend pas en charge pas les profils analogiques S7-4.
- Une voie AS-Interface par module
- Repérage automatique de l'esclave avec le repère 0
- Gestion des profils et paramètres
- Protection contre l'inversion de polarité sur les entrées de bus

Le bus AS-Interface permet alors :

- jusqu'à 31 esclaves de type repérage standard et 62 de type repérage étendu,
- jusqu'à 248 entrées et 186 sorties,
- jusqu'à 7 esclaves analogiques (4 E/S max. par esclave),
- un temps de cycle de 10 ms maximum.

Deux modules maîtres AS-Interface maximum peuvent être connectés sur un automate modulaire Twido, un automate compact TWDLC•A24DRF ou TWDLC••40DRF.

Description fonctionnelle générale

Présentation générale

Pour la configuration AS-Interface, TwidoSuite permet à l'utilisateur :

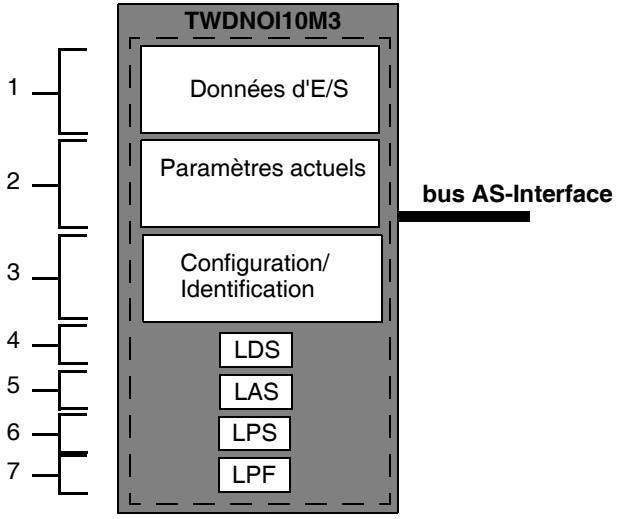
- de configurer le bus (déclaration des esclaves et attribution des adresses sur le bus) de façon manuelle,
- d'adapter la configuration par rapport à ce qui est présent sur le bus,
- de prendre en compte les paramètres des esclaves,
- de contrôler l'état du bus.

Pour cela toutes les informations en provenance ou à destination du maître AS-Interface sont stockées dans des objets (mots et bits) spécifiques.

Structure du maître AS-Interface

Le module AS-Interface intègre des champs de données qui permettent de gérer des listes d'esclaves et les images des données d'entrées / sorties. Ces informations sont stockées en mémoire volatile.

La figure ci-dessous présente l'architecture du module **TWDNOI10M3**.



Légende :

Repère	Elément	Description
1	Données d'E/S (IDI, ODI)	Images des 248 entrées et des 186 sorties du Bus AS-Interface V2.
2	Paramètres actuels (PI, PP)	Image des paramètres de tous les esclaves.
3	Configuration/Identification (CDI, PCD)	Ce champ contient tous les codes E/S et les codes identification de tous les esclaves détectés.
4	LDS	Liste de tous les esclaves détectés sur le bus.
5	LAS	Liste des esclaves activés sur le bus.
6	LPS	Liste des esclaves prévus sur le bus et configurés par TwidoSuite.
7	LPF	Liste des esclaves ayant un défaut périphérique.

Structure des équipements esclaves

Les esclaves en adressage standard disposent chacun de :

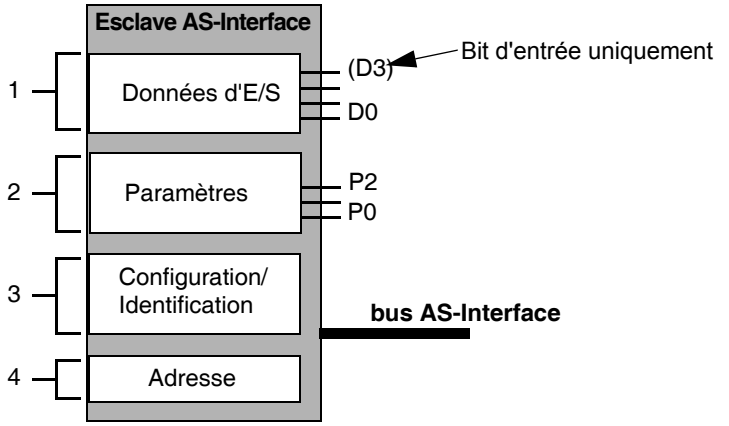
- 4 bits d'entrée/sortie,
- 4 bits de paramétrage.

Les esclaves en adressage étendu disposent chacun de :

- 4 bits d'entrée/sortie (dernier bit réservé à l'entrée uniquement),
- 3 bits de paramétrage.

Chaque esclave possède sa propre adresse, ainsi qu'un profil et sous-profil (définition de l'échange des variables).

La figure ci-dessous présente la structure d'un esclave en adressage étendu :



Légende :

Repère	Elément	Description
1	Données d'entrées/sorties	Les données d'entrées sont mémorisées par l'esclave et mises à la disposition du maître AS-Interface. Les données de sorties sont mises à jour par le module maître.
2	Paramètres	Les paramètres permettent le pilotage et la commutation des modes de marche internes au capteur ou actionneur.
3	Configuration/Identification	Ce champ contient : ● le code correspondant à la configuration des entrées/sorties (I/O), ● le code d'identification de l'esclave (ID), ● les sous-codes d'identification de l'esclave (ID1 et ID2).
4	Adresse	Adresse physique de l'esclave.
Remarque : Les paramètres de fonctionnement, adresse, données de configuration et d'identification sont sauvegardés dans une mémoire non volatile.		

Principes de mise en œuvre logicielle

Présentation

Pour respecter la philosophie adoptée dans TwidoSuite, l'utilisateur doit procéder par étapes pour créer une application AS-Interface.

Principe de mise en œuvre

L'utilisateur doit savoir comment configurer de façon fonctionnelle son bus AS-Interface (voir *p. 243*).

Le tableau ci-dessous présente les différentes phases de mise en œuvre logicielle du bus AS-Interface V2.

Mode	Phase	Description
Local	Déclaration du module	Choix de l'emplacement du module maître AS-Interface TWDNOI10M3 sur le bus d'expansion.
	Configuration de la voie du module	Choix des modes "maître".
	Déclaration des équipements esclaves	Choix pour chaque équipement : ● de son numéro d'emplacement sur le bus, ● du type d'esclave adressage standard ou adressage étendu.
	Validation des paramètres de configuration	Validation au niveau esclave.
	Validation globale de l'application	Validation de niveau application.
Local ou connecté	Symbolisation (optionnel)	Symbolisation des variables associées aux équipements esclaves.
	Programmation	Programmation de la fonction AS-Interface V2.
Connecté	Transfert	Transfert de l'application dans l'automate.
	Mise au point	Mise au point de l'application à l'aide : ● de la fenêtre AS-Interface permettant d'une part la visualisation des esclaves (adresse, paramètres), et d'autre part l'adressage des esclaves aux adresses souhaitées ; ● des écrans de diagnostic permettant d'identifier les défauts.

Note : La déclaration et la suppression du module maître AS-Interface sur le bus d'expansion se déroule comme pour un autre module d'expansion. Mais une fois deux modules maître AS-Interface déclarés sur le bus d'expansion, TwidoSuite ne permet plus d'en déclarer un autre.

Précautions avant la connexion

Avant de connecter (de façon logicielle) le PC à l'automate et pour éviter tout problème de détection :

- assurez-vous qu'il n'y a pas d'esclave présent physiquement sur le bus à l'adresse 0,
- assurez-vous qu'il n'y a pas 2 esclaves présents physiquement à la même adresse.

Description de l'écran de configuration du bus AS-Interface

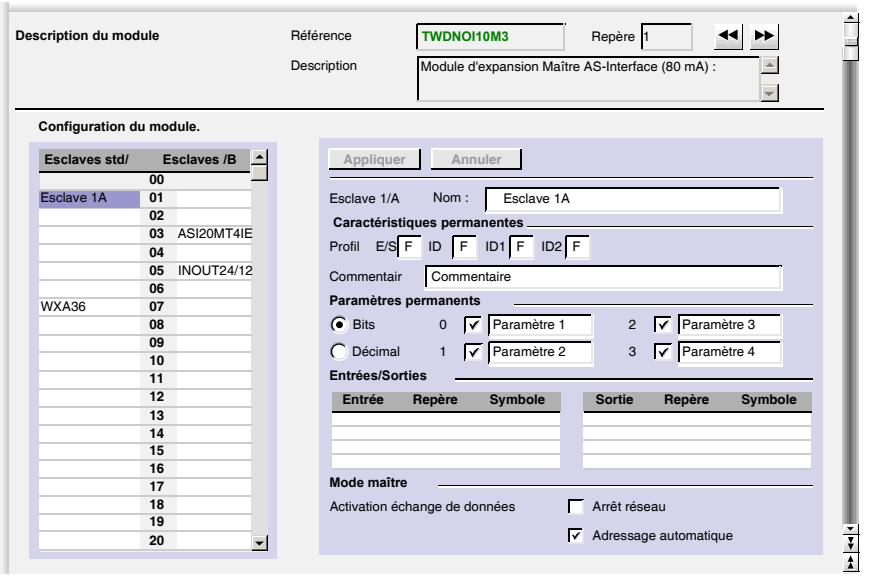
Présentation

L'écran de configuration du module maître AS-Interface donne accès aux paramètres associés au module et aux équipements esclaves.

Il permet la visualisation et la modification des paramètres en mode local.

Illustration en mode local

Illustration de l'écran de configuration en mode local :



Description de l'écran en mode local

Cet écran regroupe la totalité des informations constituant le bus en proposant trois blocs d'informations :

Blocs	Description
Configuration AS-interface	Image du bus souhaitée par l'utilisateur : visualisation des esclaves à adressage standard et étendu projetés (prévus) sur le bus. Il faut descendre le curseur de la barre verticale pour accéder aux adresses suivantes. Les adresses inaccessibles correspondent aux adresses qui ne sont pas disponibles pour la configuration d'un esclave. Par exemple, si un nouvel esclave standard est déclaré à l'adresse 1A, l'adresse 1B est alors automatiquement rendue indisponible.
Esclave xxA/B	Configuration de l'esclave sélectionné : ● Caractéristiques : codes IO, ID, ID1 et ID2 (profiles), et commentaire sur l'esclave, ● Paramètres : liste des paramètres (modifiables), sous forme binaire (4 cases à cocher) ou décimale (1 case) au choix de l'utilisateur, ● Entrées/Sorties : liste des Entrées/Sorties disponibles, et leur repère (adresse).
Mode maître	Activation ou désactivation possible des deux fonctionnalités disponibles pour ce module AS-Interface (comme par exemple l'adressage automatique). La fonction "Arrêt du réseau" vous permet de forcer le bus AS-Interface pour entrer en mode local. Le mode "Adressage automatique" est coché par défaut. Remarque : La fonction "Activation échange de données" n'est pas encore disponible.

L'écran propose également 2 boutons :

Boutons	Description
Appliquer	Permet de sauvegarder les données de configuration actuelles du bus AS-Interface. La configuration peut alors être transférée vers l'automate Twido.
Annuler	Permet d'ignorer toutes les modifications en cours.

Note : Les modifications dans l'écran de configuration ne sont possibles qu'en mode local.

Configuration du bus AS-Interface

Présentation

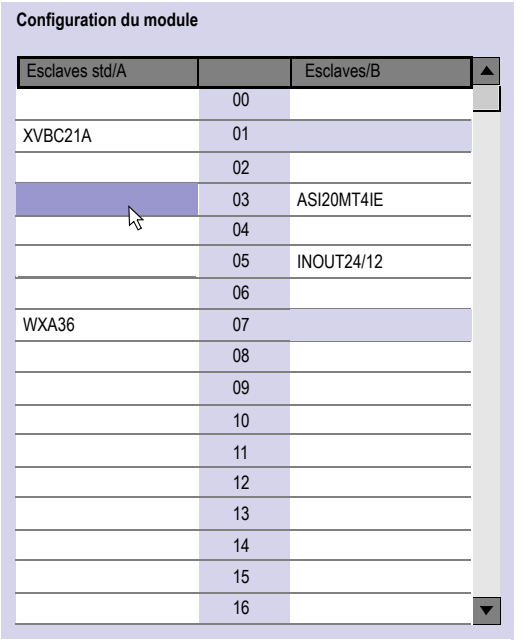
La configuration du bus AS-Interface s'effectue dans l'écran de configuration en mode local.

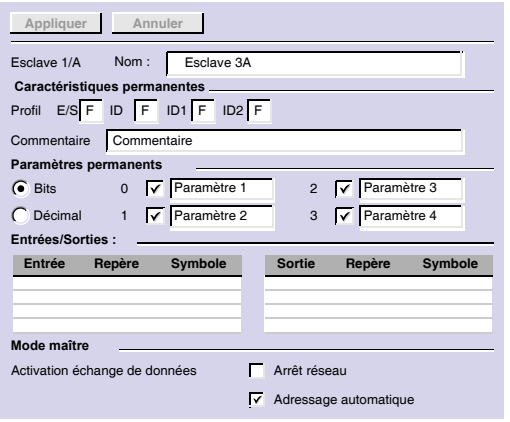
Une fois le maître AS-Interface et les modes maître sélectionnés, la configuration du bus AS-Interface consiste à configurer les équipements esclaves.

Procédure de déclaration et configuration d'un esclave

Marche à suivre pour créer ou modifier un esclave sur le bus AS-Interface V2 :

Etape	Action
1	Sélectionnez la cellule de l'adresse souhaitée (parmi celles qui sont disponibles) dans l'image du bus :



Etape	Action
2	Dans l'écran de configuration de l'esclave, saisissez ou modifiez : ● le nom du nouveau profil (limité à 13 caractères) ; ● un commentaire (optionnel). Vous pouvez également cliquer sur le bouton **Insérer depuis le catalogue** dans la barre des fonctions et sélectionnez un esclave dans la famille de profils AS-Interface préconfigurés. Illustration d'un écran de configuration pour un esclave :  Remarque : ● Dans le cas d'un nouvel esclave, un nouvel écran pour configurer l'esclave apparaît, le champ "Adresse" indique l'adresse sélectionnée, les champs de "Profil" sont à F par défaut et tous les autres champs de l'écran sont vides. ● Dans le cas d'une modification, l'écran de configuration de l'esclave apparaît avec les champs contenant les valeurs préalablement définies de l'esclave sélectionné.
3	Saisissez : ● le code IO (correspond à la configuration entrée/sortie) ; ● le code ID (identificateur), plus ID1 et ID2 pour un type étendu. Remarque : Les champs "Entrées" et "Sorties" indiquent le nombre de voie d'entrée et de sortie. Ils sont implémentés automatiquement lors de la saisie du code IO.
4	Définissez pour chaque paramètre : ● sa prise en compte par le système (case cochée en vue "Bits" ou valeur décimale entre 0 et 15 en vue "Décimal"), ● un libellé plus significatif que "Paramètre X" (optionnel). Remarque : Les paramètres sélectionnés sont l'image des paramètres permanents à fournir au maître AS-Interface.

Etape	Action
5	Modifiez "Adresse" si nécessaire (dans la limite des adresses disponibles sur le bus), en cliquant sur les flèches haut/bas à gauche de l'adresse (accès alors aux adresses autorisées) ou en saisissant directement l'adresse au clavier.
6	Validez la configuration de l'esclave en cliquant sur le bouton **Appliquer**. Le résultat est la vérification que : ● les codes IO et ID sont autorisés ; ● l'adresse de l'esclave est autorisée (en cas de saisie clavier) selon le code ID (les esclaves "banque" /B sont seulement disponibles si le code ID est égal à A). En cas d'erreur, un message avertit l'utilisateur du type d'erreur (exemple : "L'esclave ne peut avoir cette adresse") et l'écran est réaffiché avec les valeurs initiales (dans profil ou adresse selon l'erreur).

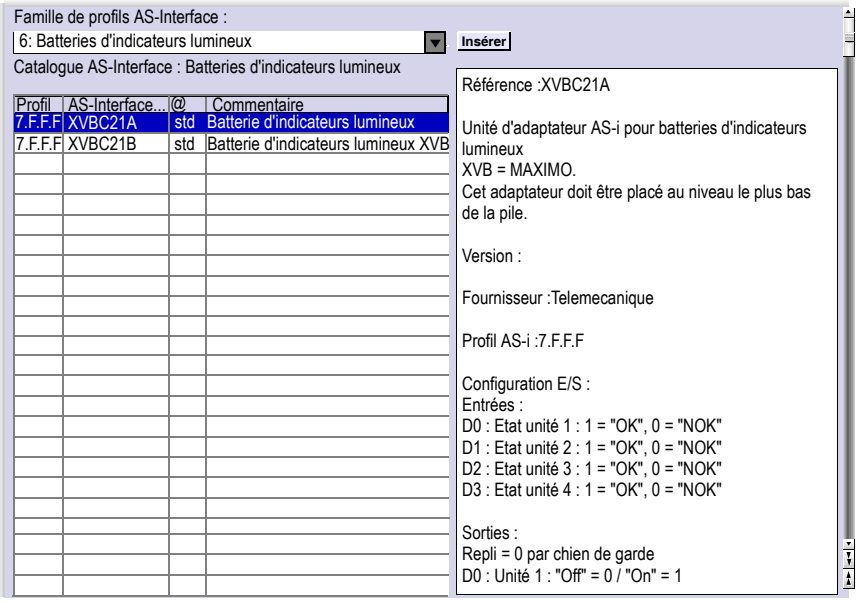
Note : Le logiciel limite le nombre de déclarations d'esclave analogique à 7.

Note : A propos du catalogue Schneider AS-Interface : lorsque vous cliquez sur le bouton Insérer depuis le catalogue, vous pouvez créer et configurer des esclaves dans "Famille privée" (autre que ceux du catalogue Schneider AS-Interface).

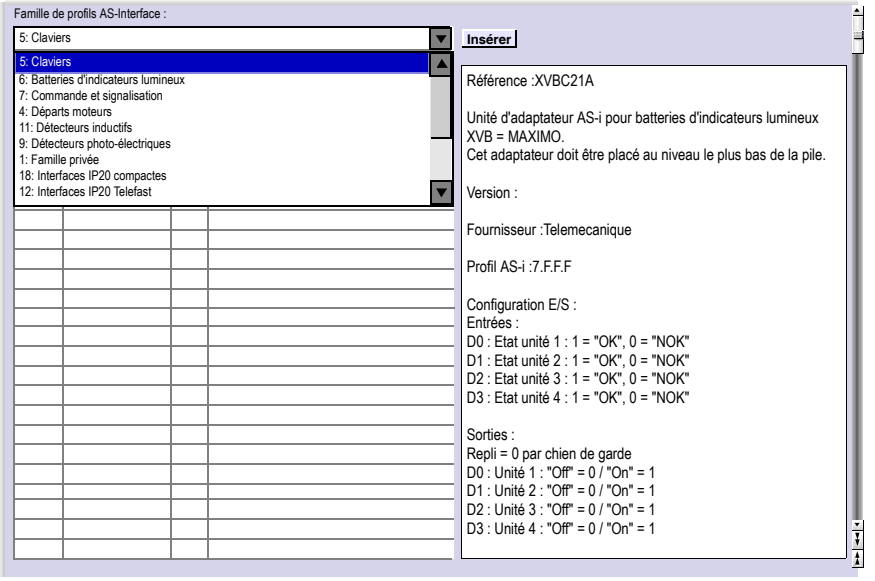
Catalogue AS-Interface

Le bouton **Insérer depuis le catalogue** permet de faciliter la configuration des esclaves sur le bus. Lorsque vous utilisez un esclave de la famille de produits Schneider, utilisez ce bouton pour simplifier et accélérer la configuration.

Lorsque vous cliquez sur le bouton **Insérer depuis le catalogue**, le volet ci-après s'ouvre :


Famille de profils AS-Interface :
6: Batteries d'indicateurs lumineux
Insérer
Catalogue AS-Interface : Batteries d'indicateurs lumineux
Profil
AS-Interface...
@
Commentaire
7.F.F.F
XVBC21A
std
Batterie d'indicateurs lumineux
7.F.F.F
XVBC21B
std
Batterie d'indicateurs lumineux XVB
Référence :XVBC21A
Unité d'adaptateur AS-i pour batteries d'indicateurs lumineux
XVB = MAXIMO.
Cet adaptateur doit être placé au niveau le plus bas de la pile.
Version :
Fournisseur :Telemecanique
Profil AS-i :7.F.F.F
Configuration E/S :
Entrées :
D0 : Etat unité 1 : 1 = "OK", 0 = "NOK"
D1 : Etat unité 2 : 1 = "OK", 0 = "NOK"
D2 : Etat unité 3 : 1 = "OK", 0 = "NOK"
D3 : Etat unité 4 : 1 = "OK", 0 = "NOK"
Sorties :
Repli = 0 par chien de garde
D0 : Unité 1 : "Off" = 0 / "On" = 1

Dans le menu déroulant, vous avez accès à toutes les familles de produits du catalogue AS-Interface Schneider :

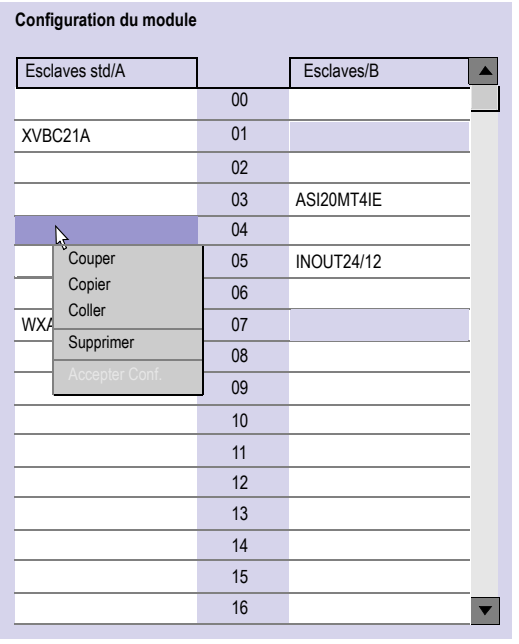

Famille de profils AS-Interface :
5: Claviers
5: Claviers
6: Batteries d'indicateurs lumineux
7: Commande et signalisation
4: Départs moteurs
11: Détecteurs inductifs
9: Détecteurs photo-électriques
1: Famille privée
18: Interfaces IP20 compactes
12: Interfaces IP20 Telefast
Insérer
Référence :XVBC21A
Unité d'adaptateur AS-i pour batteries d'indicateurs lumineux XVB = MAXIMO.
Cet adaptateur doit être placé au niveau le plus bas de la pile.
Version :
Fournisseur :Telemecanique
Profil AS-i :7.F.F.F
Configuration E/S :
Entrées :
D0 : Etat unité 1 : 1 = "OK", 0 = "NOK"
D1 : Etat unité 2 : 1 = "OK", 0 = "NOK"
D2 : Etat unité 3 : 1 = "OK", 0 = "NOK"
D3 : Etat unité 4 : 1 = "OK", 0 = "NOK"
Sorties :
Repli = 0 par chien de garde
D0 : Unité 1 : "Off" = 0 / "On" = 1
D1 : Unité 2 : "Off" = 0 / "On" = 1
D2 : Unité 3 : "Off" = 0 / "On" = 1
D3 : Unité 4 : "Off" = 0 / "On" = 1

Lorsque vous avez choisi un produit, la liste des esclaves correspondants s'affiche. Cliquez sur l'esclave souhaité et validez en cliquant sur "Insérer".

Note :

- Cliquez sur un nom de produit dans le catalogue AS-Interface pour afficher ses caractéristiques dans le volet de droite.
- Vous pouvez rajouter et configurer des esclaves qui ne font pas partie du catalogue Schneider. Il vous suffit de choisir la famille privée et de configurer ce nouvel esclave.

Menu contextuel Un menu contextuel s'affiche lorsque vous cliquez avec le bouton droit :



Le menu contextuel comporte les fonctions suivantes :

- Couper (Ctrl+X)
- Copier (Ctrl+C)
- Coller (Ctrl+V)
- Supprimer (Suppr)

Description de la fenêtre AS-Interface en mode connecté

Présentation

Lorsque le PC est **connecté** à l'automate (après le chargement de l'application vers l'automate), la fenêtre AS-Interface affiche les caractéristiques du mode connecté.

En mode connecté, la fenêtre AS-Interface affiche une image dynamique du bus physique avec les éléments suivants :

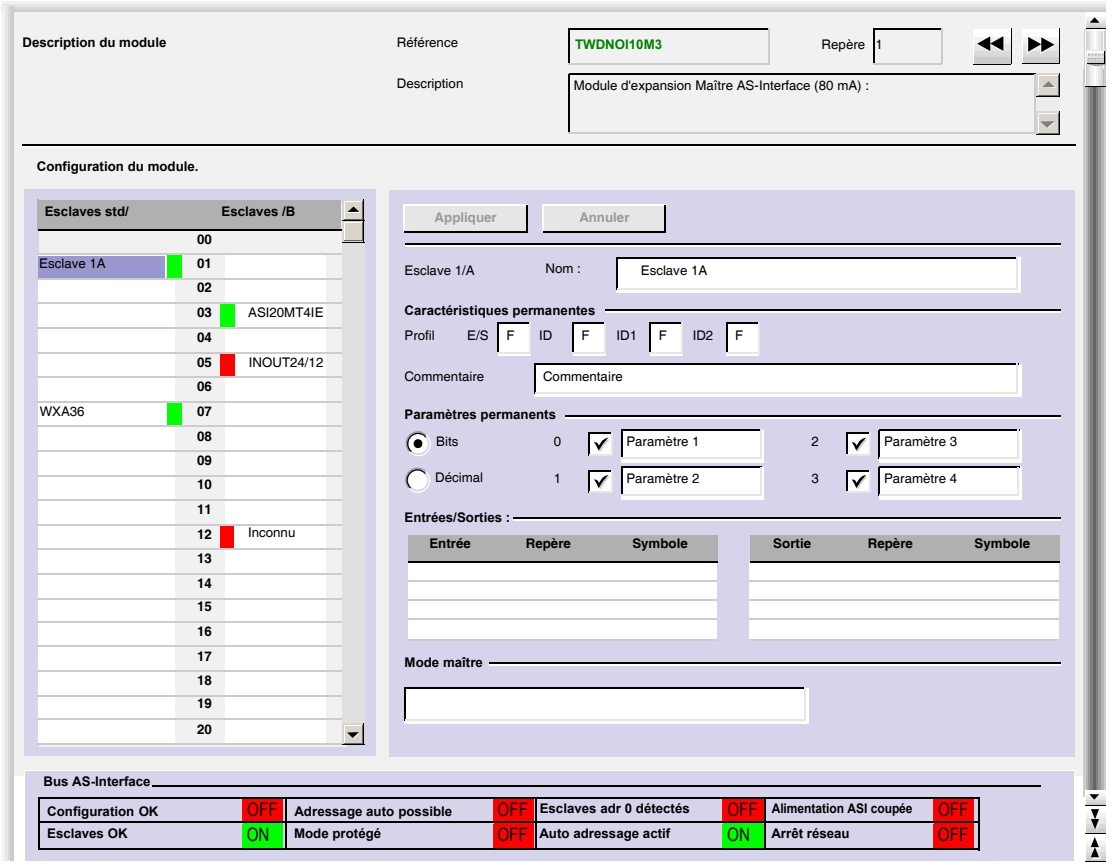
- la liste des esclaves prévus (saisis) pendant la configuration avec leur nom, et la liste des esclaves détectés (de nom inconnu si non prévus),
- l'état du coupleur AS-Interface et des équipements esclaves,
- l'image du profil, des paramètres et des valeurs des entrées/sorties des esclaves sélectionnés.

Il permet également à l'utilisateur :

- d'obtenir un diagnostic des esclaves en erreur (voir *p. 235*),
- de modifier l'adresse d'un esclave en mode connecté (voir *p. 236*),
- de transmettre l'image des esclaves à l'écran de configuration (voir *p. 238*),
- d'adresser tous les esclaves aux adresses souhaitées (lors de la première mise au point).

Illustration de la fenêtre AS-Interface

L'illustration de la fenêtre AS-Interface (en mode connecté uniquement) se présente de la manière suivante :



Description de la fenêtre AS-Interface

La fenêtre AS-Interface indique les mêmes informations que l'écran de configuration (voir *p. 226*).

Les différences sont listées dans le tableau suivant :

Bloc	Description
Configuration AS-interface V2	Image du bus physique. Inclut l'état des esclaves : ● voyant vert : l'esclave à cette adresse est actif. ● voyant rouge : l'esclave à cette adresse est en erreur, et un message informe du type d'erreur dans la fenêtre "Erreur sur le réseau".
Esclave xxA/B	Image de la configuration de l'esclave sélectionné : ● Caractéristiques : image du profil détecté (non disponible, non modifiable), ● Paramètres : image des paramètres détectés. L'utilisateur peut uniquement choisir le format d'affichage des paramètres, ● Entrées/Sorties : les valeurs des entrées/sorties détectées sont affichées, non modifiables.
Erreur sur le réseau	Informe du type d'erreur si l'esclave sélectionné est en erreur.
Bus AS-Interface	Informations résultantes d'une commande implicite "Read Status". ● indique l'état du bus : par exemple "Configuration OK = OFF" indique que la configuration prévue par l'utilisateur ne correspond à la configuration physique du bus, ● indique les fonctionnalités autorisées au module maître AS-Interface : par exemple "Adressage auto actif = ON" indique que le mode maître adressage automatique est autorisé.

Visualisation des états des esclaves

Lorsque le voyant associé à une adresse est rouge, l'esclave associé à cette adresse est en erreur. La fenêtre "Erreur sur le réseau" fournit alors le diagnostic de l'esclave sélectionné.

Descriptif des erreurs :

- le profil prévu par l'utilisateur en configuration à une adresse donnée, ne correspond pas au profil réel détecté à cette adresse sur le bus (diagnostic : Erreur de profil"),
- un nouvel esclave non prévu en configuration, est détecté sur le bus : un voyant rouge est alors affiché pour cette adresse et le nom de l'esclave affiché est "Inconnu" (diagnostic : "Esclave non projeté"),
- défaut périphérique si l'esclave détecté le supporte (diagnostic : "Défaut périphérique"),
- un profil est prévu en configuration mais aucun esclave est détecté à cette adresse sur le bus (diagnostic : "Esclave non détecté").

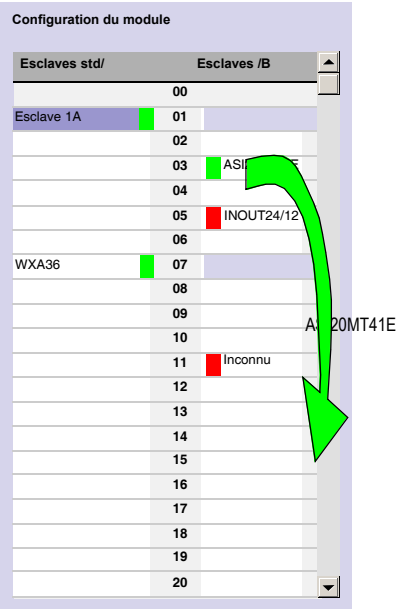
Modification de l'adresse d'un esclave

Présentation

Dans la fenêtre AS-Interface, l'utilisateur peut modifier l'adresse d'un esclave en mode connecté.

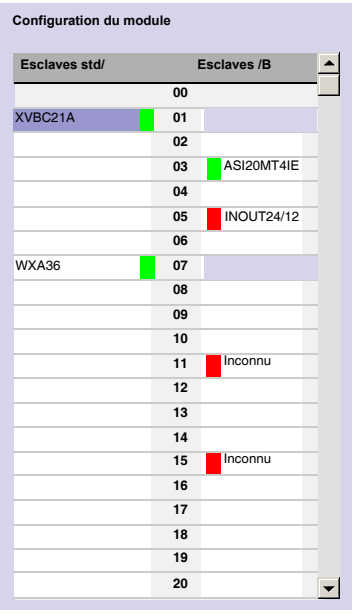
Modification de l'adresse d'un esclave

Le tableau suivant présente la procédure pour modifier l'adresse d'un esclave :

Etape	Description
1	Accédez à la fenêtre AS-Interface.
2	Sélectionnez un esclave dans la zone "Configuration AS-interface V2".
3	Exécutez un "glisser et déposer" à l'aide de la souris vers la cellule correspondant à l'adresse désirée. Illustration : glisser-déposer de l'esclave 3B vers l'adresse 15B 

Etape	Description

Résultat :
Un contrôle de tous les paramètres de l'esclave s'effectue automatiquement pour vérifier si l'opération est possible.
Illustration du résultat :



Après l'opération, le diagnostic de l'esclave à l'adresse 3B affiche "esclave non détecté" indiquant que l'esclave prévu à cette adresse n'est plus présent. En sélectionnant l'adresse 15B, on retrouve bien le profil et les paramètres de l'esclave déplacé, le nom de l'esclave reste, quant à lui, inconnu car il n'était pas prévu à cette adresse là.

Note : Le profil et les paramètres d'un esclave ne sont pas attachés à son nom. Plusieurs esclaves de noms différents peuvent avoir les mêmes profils et paramètres.

Mise à jour de la configuration du bus AS-Interface en mode connecté

Présentation

En mode connecté, aucune modification de l'écran de configuration n'est autorisée et la configuration physique et la configuration logicielle peuvent être différentes. Toute différence de profil ou paramètres d'un esclave prévu ou non en configuration peut être prise en compte dans l'écran de configuration, il est en effet possible de transmettre toute modification à l'écran de configuration avant de transférer la nouvelle application vers l'automate.

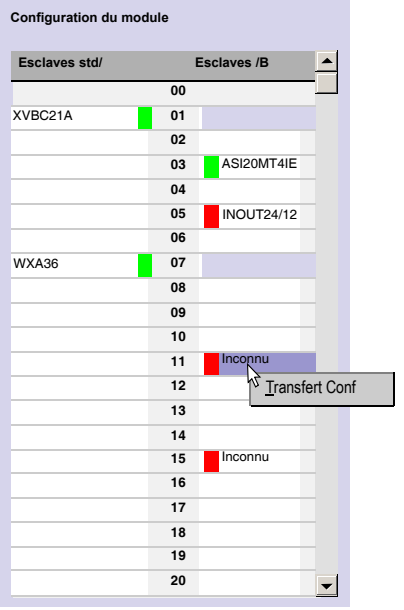
La procédure à suivre pour prendre en compte la configuration physique, est la suivante :

Etape	Désignation
1	Transfert de la configuration de l'esclave désiré vers l'écran de configuration.
2	Acceptation de la configuration dans l'écran de configuration.
3	Validation de la nouvelle configuration.
4	Transfert de l'application au module.

Transfert de l'image d'un esclave vers la configuration

Dans le cas de la détection d'un esclave sur le bus non prévu dans la configuration, un esclave "Inconnu" apparaît dans la zone "Configuration AS-interface V2" de la fenêtre AS-Interface à l'adresse détectée.

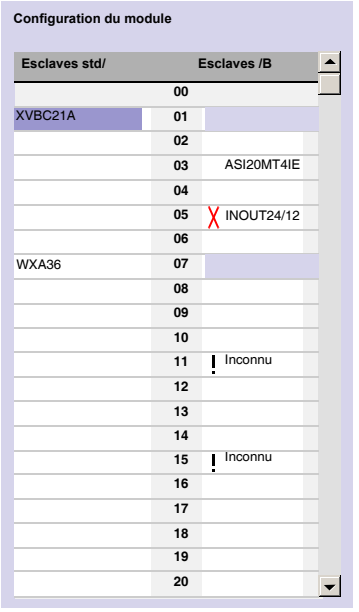
Le tableau suivant indique la procédure pour transférer l'image de l'esclave "Inconnu" dans l'écran de configuration :

Etape	Description
1	Accédez à la fenêtre AS-Interface.
2	Sélectionnez l'esclave désiré dans la zone "Configuration AS-interface V2".
3	Exécutez un clic droit sur la souris pour choisir "Transfert Conf". Illustration :  Résultat : L'image de l'esclave sélectionné (image du profil et paramètres) est alors transféré à l'écran de configuration.
4	Recommencez l'opération pour chacun des esclaves dont on veut transférer l'image vers l'écran de configuration.

Retour à l'écran de configuration

Quand l'utilisateur revient dans l'écran de configuration, tous les nouveaux esclaves (non prévus) transférés sont visibles.

Illustration de l'écran de configuration après le transfert de tous les esclaves :



Légende :

- la croix signifie qu'il y a des différences entre l'image du profil de l'esclave transféré, et le profil souhaité initialement dans l'écran de configuration.
- Le point d'exclamation signifie qu'un nouveau profil a été introduit dans l'écran de configuration.

Explication :

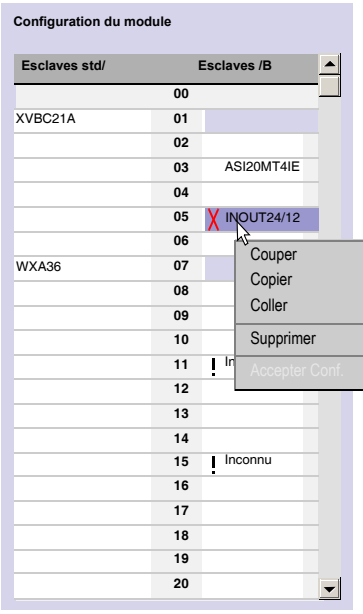
L'écran de configuration présente toujours l'image permanente de la configuration souhaitée (d'où la présence de l'esclave en 3B malgré le changement d'adresse (voir *236*)), complétée de l'image courante du bus.

Les profils et paramètres des esclaves prévus qui sont affichés correspondent à ceux qui étaient prévus. Les profils et paramètres des esclaves inconnus qui sont affichés correspondent aux images de ceux détectés.

Procédure pour le transfert de l'application définitive vers le module

Avant de transférer une nouvelle application vers le module, l'utilisateur peut pour chacun des esclaves prendre en compte l'image du profil et des paramètres détectée (transférée à l'écran de configuration) ou modifier la configuration "à la main" (voir *p. 227*).

Le tableau suivant décrit la marche à suivre pour la validation et le transfert de la configuration définitive vers le module :

Etape	Action
1	Déconnectez de façon logicielle le PC du module. Remarque : Aucune modification dans l'écran de configuration n'est possible si le PC est connecté au module.
2	Faites un clic droit souris sur l'esclave souhaité.
3	2 choix : ● choisissez "Accepter Conf" pour accepter le profil **détecté** de l'esclave sélectionné. Illustration :  Pour chacun des esclaves marqués d'une croix, un message avertit l'utilisateur que cette opération écrasera le profil initial (affiché dans l'écran) de l'esclave. ● Choisissez les autres choix du menu contextuel pour configurer à la main l'esclave sélectionné.
4	Recommencez l'opération pour chacun des esclaves désirés dans la configuration.
5	Appuyez sur le bouton "OK" pour valider et créer la nouvelle application. Résultat : Retour automatique à l'écran principal.
6	Transférez l'application vers le module.

Adressage automatique d'un esclave AS-Interface V2

Présentation

Chaque esclave présent sur le bus AS-Interface doit se voir affecter (par configuration) une adresse physique unique. Celle-ci doit être l'image de celle déclarée dans TwidoSuite.

TwidoSuite offre un service d'adressage automatique des esclaves qui évite ainsi d'utiliser une console AS-Interface.

Le service d'adressage automatique est utilisé pour :

- remplacer un esclave défaillant,
- insérer un nouvel esclave.

Marche à suivre

Le tableau ci-dessous présente la marche à suivre pour définir le paramètre **Adressage automatique**.

Etape	Action
1	Accédez à l'écran de configuration du module maître AS-Interface V2.
2	Cliquez sur la case à cocher **Adressage automatique** située dans la zone **Mode maître**. **Résultat** : Le service **Adressage automatique** sera actif (case cochée) ou non actif (case non cochée). **Remarque** : Le paramètre **Adressage automatique** est sélectionné par défaut dans l'écran de configuration.

Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante

Présentation

Il est possible d'insérer un équipement dans une configuration AS-Interface V2 existante sans avoir recours à l'utilisation du programmateur de poche.

Cette opération est possible dès lors que :

- le service **Adressage automatique** du mode de configuration est actif (voir *p. 242*),
- un seul esclave est absent dans la configuration physique,
- l'esclave à insérer est prévu dans l'écran de configuration,
- l'esclave possède le profil attendu par la configuration,
- l'esclave possède l'adresse 0(A).

Ainsi, le module AS-Interface V2 affectera automatiquement à l'esclave la valeur prédéfinie dans la configuration.

Marche à suivre

Le tableau ci-dessous présente la marche à suivre pour que l'insertion automatique d'un nouvel esclave soit effective.

Etape	Action
1	Ajoutez le nouvel esclave dans l'écran de configuration en mode local.
2	Faites un transfert de configuration vers l'automate en mode connecté.
3	Raccordez physiquement le nouvel esclave d'adresse 0(A) sur le bus AS-Interface V2.

Note : Il est possible de modifier une application en réalisant la manipulation ci-dessus autant de fois que nécessaire.

Remplacement automatique d'un esclave AS-Interface V2 défectueux

Principe

Lorsqu'un esclave est déclaré en défaut, il est possible de le remplacer de façon automatique par un esclave de même type.

Le remplacement s'effectue sans arrêt du bus AS-Interface V2 et sans manipulation particulière dès lors que le service **Adressage automatique** du mode de configuration est actif (voir *p. 242*).

Deux possibilités peuvent se présenter :

- l'esclave venant en remplacement est programmé avec la même adresse à l'aide du programmateur de poche et possède le même profil et sous-profil que l'esclave défectueux. Il sera donc inséré automatiquement dans la liste des esclaves détectés (LDS) et dans la liste des esclaves actifs (LAS),
- l'esclave venant en remplacement est vierge (adresse 0(A), esclave neuf) et possède le même profil que l'esclave défectueux. Il prendra automatiquement l'adresse de l'esclave remplacé et sera donc inséré dans la liste des esclaves détectés (LDS) et dans la liste des esclaves actifs (LAS).

Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2

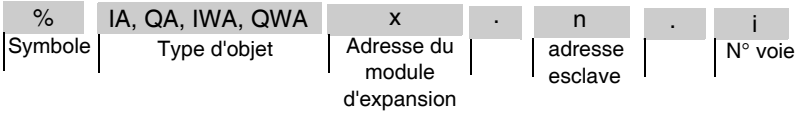
Présentation

Cette page présente les spécificités de l'adressage des entrées/sorties TOR ou analogiques des équipements esclaves.

Pour éviter toute confusion avec les E/S déportés, de nouveaux symboles sont proposés avec une syntaxe AS-Interface : %I**A** par exemple.

Illustration

Rappel des principes de repérage :



Valeurs spécifiques

Le tableau ci-dessous donne les valeurs spécifiques aux objets des esclaves AS-Interface V2 :

Elément	Valeurs	Commentaire
IA	-	Image de la sortie physique TOR de l'esclave.
QA	-	Image de la sortie physique TOR de l'esclave.
IWA	-	Image de l'entrée physique analogique de l'esclave.
QWA	-	Image de la sortie physique analogique de l'esclave.
x	1 à 7	Adresse du module AS-Interface sur le bus d'expansion
n	0A à 31B	L'emplacement 0 n'est pas configurable.
i	0 à 3	-

Exemples

Le tableau présente quelques exemples d'adressage des E/S :

Objet d'E/S	Description
%IWA4.1A.0	entrée analogique 0 de l'esclave 1A du module AS-Interface positionné en 4 sur le bus d'expansion.
%QA2.5B.1	sortie TOR 1 de l'esclave 5B du module AS-Interface positionné en 2 sur le bus d'expansion.
%IA1.12A.2	entrée TOR 2 de l'esclave 12A du module AS-Interface positionné en 1 sur le bus d'expansion.

Echanges implicites

Les objets décrits ci-dessus sont échangés de façon implicite, c'est à dire qu'ils sont échangés de façon automatique à chaque cycle automate.

Programmation et diagnostic du bus AS-Interface V2

Echanges explicites

Des objets (mots et bits) associés au bus AS-Interface apportent des informations (ex. : fonctionnement du bus, état des esclaves...) et des commandes supplémentaires pour effectuer une programmation avancée de la fonction AS-Interface.

Ces objets sont échangés de façon explicite entre l'automate Twido et le maître AS-Interface par le bus d'expansion :

- sur demande du programme utilisateur à l'aide de l'instruction : ASI_CMD (voir plus bas "Présentation de l'instruction ASI_CMD"),
- Via la fenêtre AS-Interface ou la table d'animation.

Mots systèmes spécifiques réservés

Les mots systèmes réservés dans l'automate Twido pour les modules maîtres AS-Interface permettent de connaître l'état du réseau : %SW73 est réservé pour le premier module d'expansion AS-Interface, et %SW74 pour le second. Seuls les 5 premiers bits de ces mots sont utilisés, ils sont en lecture seule.

Le tableau suivant présente les bits utilisés :

Mots système	Bit	Description
%SW73 et %SW74	0	état du système (= 1 si configuration OK, 0 sinon)
	1	échange de données (= 1 si échange de données activée, 0 si en mode Data Exchange Off (voir *p. 250*))
	2	système en stop (= 1 si le mode Offline (voir *p. 250*) est activé, 0 sinon)
	3	instruction ASI_CMD terminée (= 1 si terminée, 0 si en cours)
	4	erreur instruction ASI_CMD (= 1 si erreur dans instruction, 0 sinon)

Exemple d'utilisation (pour le premier module d'expansion AS-Interface) :

Avant d'utiliser une instruction ASI_CMD, le bit %SW73:X3 doit être vérifié pour savoir si une instruction n'est pas en cours : vérifier que %SW73:X3 = 1.

Pour savoir, si l'instruction s'est ensuite bien exécutée, vérifier que le bit %SW73:X4 est égal à 0.

Présentation de l'instruction ASI_CMD

Par programme utilisateur, l'instruction ASI_CMD permet à l'utilisateur de programmer son réseau et d'obtenir le diagnostic des esclaves. Les paramètres de l'instruction sont transmis par mots internes (mots mémoire) **%MW**x.

La syntaxe de l'instruction est la suivante :

ASI_CMD`n` **%MW**`x`**:**`l`

Légende :

Symbole	Désignation
n	Adresse du module d'expansion AS-Interface (1 à 7)
x	Numéro du premier mot interne (mémoire) passé en paramètre
l	Longueur de l'instruction en nombre de mots (2)

Utilisation de l'instruction ASI_CMD

Le tableau suivant décrit l'action de l'instruction ASI_CMD en fonction de la valeur des paramètres %MW(x), et %MW(x+1) quand nécessaire. Pour les demandes de diagnostic des esclaves, le résultat est retourné dans %MW(x+1).

%MWx	%MWx+1	Action
1	0	quitte le mode Offline.
1	1	passe en mode Offline.
2	0	interdit l'échange de données entre le maître et ses esclaves (entre dans le mode Data Exchange Off).
2	1	autorise l'échange de données entre le maître et ses esclaves (sort du mode Data Exchange Off).
3	réservé	-
4	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 0A à 15A (1 bit par esclave).
5	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 16A à 31A (1 bit par esclave).
6	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 0B à 15B (1 bit par esclave).
7	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 16B à 31B (1 bit par esclave).
8	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 0A à 15A (1 bit par esclave).
9	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 16A à 31A (1 bit par esclave).
10	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 0B à 15B (1 bit par esclave).
11	Résultat	lit la liste des esclaves détectés (table LDS)de l'adresse 16B à 31B (1 bit par esclave).
12	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 0A à 15A (1 bit par esclave).
13	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 16A à 31A (1 bit par esclave).
14	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 0B à 15B (1 bit par esclave).
15	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 16B à 31B (1 bit par esclave).
16	Résultat	lit l'état du bus. Voir détail du résultat dans le paragraphe suivant.
32	Param	écrit un nouveau paramètre dans un esclave AS-Interface (table PI). Param : ● Octet 0 : nouveau paramètre à écrire – 0 à 15 ● Octet 1 : adresse - 0 à 31 (pour 0A à 31A) et 100 à 131 (pour 0B à 31B)
33	Param	lit un paramètre sur un esclave AS-Interface (table PI). Param : ● Octet 0 : nouveau paramètre à écrire – 0 à 15 ● Octet 1 : adresse - 0 à 31 (pour 0A à 31A) et 100 à 131 (pour 0B à 31B)

Note : L'état du bus est mis à jour à chaque cycle automate. Mais le résultat de l'instruction ASI_CMD de lecture du bus n'est disponible qu'au cycle automate suivant.

Détail du résultat de l'instruction ASI_CMD pour lire l'état du bus

Dans le cas d'une lecture de l'état du bus par l'instruction ASI_CMD (valeur du paramètre %MWx égale à 16), le format du résultat dans le mot %MWx+1 est la suivante :

%MWx+1		**Désignation (1=OK, 0=NOK)**
poids faible	bit 0	Configuration OK
	bit 1	LDS.0 (esclave présent à l'adresse 0)
	bit 2	Autoadressage actif
	bit 3	Auto adressage disponible
	bit 4	Mode Configuration actif
	bit 5	Normal opération active
	bit 6	APF (problème d'alimentation)
	bit 7	Offline prêt
poids fort	bit 0	Périphérique OK
	bit 1	Echange de données actif
	bit 2	Mode Offline
	bit 3	Mode normal (1)
	bit 4	Défaut communication avec le maître AS-Interface
	bit 5	Instruction ASI_CMD en cours
	bit 6	Instruction ASI_CMD en erreur

Détail du résultat de l'instruction ASI_CMD pour lire l'état des esclaves

Dans le cas d'un diagnostic des esclaves par l'instruction ASI_CMD (valeur %MWx comprise entre 4 et 15), l'état des esclaves est retourné dans les bits (1=OK) du mot %MWx+1. Le tableau suivant donne le détail du résultat en fonction de la valeur du mot %MWx :

%MWx	**%MWx+1**															
valeur	octet poids fort								octet poids faible							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
4, 8, 12	15 A	14 A	13 A	12 A	11 A	10 A	9 A	8 A	7 A	6 A	5 A	4 A	3 A	2 A	1 A	0 A
5, 9, 13	31 A	30 A	29 A	28 A	27 A	26 A	25 A	24 A	23 A	22 A	21 A	20 A	19 A	18 A	17 A	16 A
6, 10, 14	15 B	14 B	13 B	12 B	11 B	10 B	9 B	8 B	7 B	6 B	5 B	4 B	3 B	2 B	1 B	0 B
7, 11, 15	31 B	30 B	29 B	28 B	27 B	26 B	25 B	24 B	23 B	22 B	21 B	20 B	19 B	18 B	17 B	16 B

Pour lire si l'esclave 20B est actif, l'instruction ASI_CMD doit être exécutée avec le mot interne %MWx de valeur 7. Le résultat est retourné dans le mot interne %MWx+1, l'état de l'esclave 20B est donné par la valeur du bit 4 de l'octet de poids faible : si le bit 4 est égal à 1 alors l' esclave 20B est actif.

Exemples de programmation de l'instruction ASI_CMD

Pour forcer le passage du maître AS-Interface (positionné en 1 sur le bus d'expansion) en mode Offline :

```
LD 1
[%MW0 := 16#0001]
[%MW1 := 16#0001]
LD %SW73:X3          //Si aucune instruction ASI_CMD est en cours, on continue
[ASI_CMD1 %MW0:2]    //pour forcer le passage en mode Offline
```

Pour lire la table des esclaves actifs de l'adresse 0A à 15A :

```
LD 1
[%MW0 := 16#0004]
[%MW1 := 16#0000          //optionnel]
LD %SW73:X3  //Si aucune instruction ASI_CMD est en cours, on continue
[ASI_CMD1 %MW0:2]    //pour lire la table LAS de l'adresse 0A à 15A
```

Mode de fonctionnement du module interface bus AS-Interface V2

Présentation

Le module interface bus AS-Interface TWDNOI10M3 dispose de trois modes de fonctionnement répondant chacun à des besoins particuliers. Ces modes sont :

- le mode protégé,
- le mode Offline
- le mode Data Exchange Off.

L'utilisation de l'instruction ASI_CMD (voir *p. 246*) dans un programme utilisateur permet de rentrer ou de sortir de ces modes.

Mode protégé

Le mode de fonctionnement protégé est le mode généralement utilisé pour une application en exploitation. Il implique que le module AS-Interface V2 soit configuré dans TwidoSuite. Celui-ci :

- vérifie en permanence que la liste des esclaves détectés est égale à la liste des esclaves prévus,
- surveille l'alimentation.

Dans ce mode, un esclave ne sera activé que s'il a été déclaré dans la configuration et détecté.

A la mise sous tension ou pendant la phase de configuration, l'automate Twido force le module AS-Interface en mode protégé.

Mode Offline

A l'arrivée dans le mode Offline, le module effectue d'abord une remise à zéro de tous les esclaves présents et arrête les échanges sur le bus. Pendant le mode Offline, les sorties sont forcées à zéro.

En dehors de l'usage du bouton PB2 sur le module AS-Interface TWDNOI10M3, le mode Offline est accessible de façon logicielle par l'instruction ASI_CMD (voir *p. 249*), de même pour quitter le mode et revenir au mode protégé.

Mode Data Exchange Off

A l'arrivée dans le mode Data Exchange Off, les échanges sur le bus continuent à fonctionner, mais les données ne sont plus rafraîchies.

Ce mode n'est accessible que par l'instruction ASI_CMD (voir *p. 247*).

Installation et configuration du bus de terrain CANopen

10

Présentation

Objet de ce chapitre

Ce chapitre décrit les procédures d'installation et de configuration du module maître CANopen TWDNCO1M, du bus CANopen intégré de Twido Extreme et de ses équipements esclaves sur le bus de terrain CANopen.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
10.1	Présentation du bus de terrain CANopen	252
10.2	Mise en œuvre du bus CANopen	266

10.1 Présentation du bus de terrain CANopen

Aperçu

Objet de cette section

Cette section est destinée à vous apporter des connaissances générales sur la technologie du bus de terrain CANopen et à vous présenter la terminologie CAN utilisée dans le reste du chapitre.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Base de connaissances CANopen	253
A propos de CANopen	254
Boot-Up CANOpen	257
Transmission de Process Data Object (PDO)	260
Accès aux données à l'aide d'échanges explicites (SDO)	262
"Node Guarding" et "Life Guarding"	263
Gestion du bus interne	265

Base de connaissances CANopen

Introduction

La section suivante détaille les termes techniques et acronymes liés aux communications sur un réseau CANopen.

Fichier EDS

EDS (Electronic Data Sheet, document de description électronique)

Un fichier EDS comporte une description des propriétés de communication d'un équipement d'un réseau CAN (débits, types de transmission, caractéristiques d'E/S, etc.). Il est fourni par le fabricant de l'équipement et est utilisé dans l'outil de configuration lors du paramétrage d'un nœud (par exemple, un pilote dans un système d'exploitation).

PDO

PDO (Process Data Object, objet données de traitement)

Trame CANopen contenant des données d'E/S.

On distingue :

- les Transmit-PDOs (TPDO avec des données fournies par un nœud) ;
- les Receive-PDOs (RPDO avec des données utilisées par un nœud).

Le sens de la transmission est toujours considéré par rapport à un nœud. Un PDO (TPDO ou RPDO) ne contient pas nécessairement l'image complète des données d'un nœud. En général, les données d'entrée analogique et les données d'entrée TOR sont réparties dans différents TPDO. Cette caractéristique est également valable pour les sorties.

SDO

SDO (Service Data Object, objet de données de service)

Trame CANopen contenant des paramètres.

Les SDO sont généralement utilisés pour lire ou écrire des paramètres lorsqu'une application est en cours.

COB-ID

COB-ID (Communication Object Identifier, identifiant de l'objet de communication)

Chaque trame CANopen commence par un COB-ID faisant office d'identifiant dans la trame CAN. Au cours de la phase de configuration, chacun des nœuds reçoit le COB-ID de la trame (ou des trames) dont il est le fournisseur (ou le consommateur).

A propos de CANopen

Introduction

CANopen est un protocole bus de terrain standard destiné aux systèmes d'automates industriels. Il est particulièrement adapté aux automates en temps réel, car il constitue une solution efficace et économique pour les applications industrielles mobiles et embarquées.

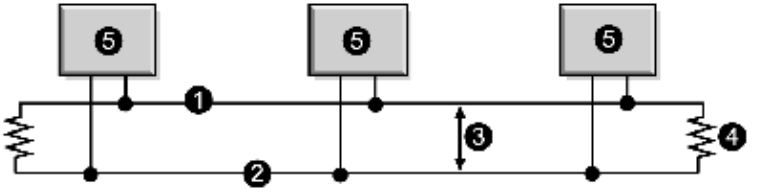
Protocole CANopen

Le protocole CANopen a été créé en tant que sous-ensemble de la couche CAL (CAN Application Layer - Couche d'application CAN). Grâce à sa fonction de définition de profils, ce protocole s'adapte au mieux aux composants industriels standard. CANopen est une norme CiA (CAN in Automation) qui a été largement adoptée par le secteur de l'automatisme dès sa mise sur le marché. CANopen est désormais reconnue comme norme européenne pour les systèmes industriels basés sur le réseau CAN.

Couche physique

La couche CAN fait appel à une ligne de bus bifilaire différentielle (retour commun). Le signal CAN correspond à la différence de tension entre les deux lignes (CAN High et CAN Low) (voir schéma ci-dessous).

Le schéma suivant présente les composants de la couche physique d'un bus CAN bifilaire :



1 Ligne CAN High (état haut)
2 Ligne CAN Low (état bas)
3 Différence potentielle entre les signaux CAN High et CAN Low
4 Résistance de 120Ω
5 Nœud

La paire de fils du bus peut être blindée, torsadée ou parallèle, tant que les exigences en matière de compatibilité électromagnétique sont respectées. Une structure composée d'une seule ligne permet de limiter la réflexion.

Profils CANopen

Profil de communication

La famille de profils CANopen s'appuie sur un "profil de communication", qui détermine les principaux mécanismes de communication et fournit leur description (DS301).

Profil des équipements

Les types d'équipement les plus utilisés dans le domaine de l'automatisation industrielle sont présentés dans les profils d'équipement (Device profiles). Ces profils décrivent également les fonctionnalités d'un équipement.

Voici quelques exemples d'équipements standard pris en charge :

- Modules d'E/S TOR et analogiques (DS401)
- Moteurs (DS402)
- Contrôleurs (DSP403)
- Automates asservis (DSP404)
- Automates (DS405)
- Codeurs (DS406)

Configuration des équipements via le bus CAN

La capacité à configurer les équipements via le bus CAN correspond à une exigence essentielle réclamée par les fabricants (pour chaque famille de profils) afin d'assurer l'autonomie du système.

Caractéristiques générales des profils CANopen

CANopen constitue un ensemble de profils destinés aux systèmes CAN comprenant les caractéristiques suivantes :

- système à bus ouvert ;
- échange de données en temps réel sans surcharge du protocole ;
- conception modulaire avec possibilité de redimensionnement ;
- interopérabilité et interchangeabilité des équipements ;
- prise en charge par un grand nombre de fabricants partout dans le monde ;
- configuration réseau normalisée ;
- accès à l'ensemble des paramètres d'équipement ;
- synchronisation et circulation de données de traitement cycliques et/ou de données d'événement (possibilité de temps de réponse système courts).

Certification des produits CANopen

La totalité des fabricants mettant sur le marché des produits certifiés CANopen est membre de l'organisation CiA. En tant que membre actif de cette organisation, Schneider Electric Industries SAS développe ses produits conformément aux spécifications publiées par cet organisme.

Normes CAN

Les spécifications CANopen sont définies par l'organisation CiA et sont accessibles (avec quelques restrictions) à partir du site : http://www.can-cia.com. Les codes sources des équipements maître et esclave sont disponibles auprès des différents fournisseurs.

Note : Pour obtenir plus d'informations sur les spécifications et les mécanismes de CANopen, consultez la page d'accueil de l'organisation CiA (http://www.can-cia.de/).

Communication sur un réseau CANopen

Le profil de communication est établi en fonctions des protocoles et services CAL (couche application basée sur le réseau CAN).

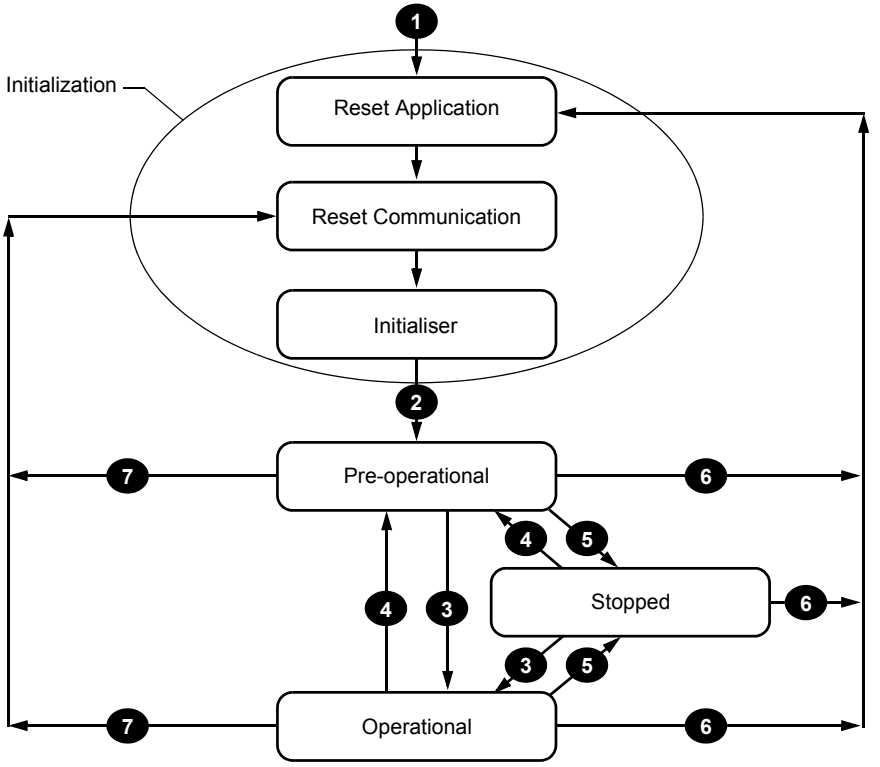
Il permet à l'utilisateur d'accéder à deux types d'échange : SDO et PDO.

Lors de la mise sous tension, l'équipement entre dans une phase d'initialisation, puis passe en mode Pré-opérationnel. A ce stade, seules les communications SDO sont autorisées. Après réception d'une instruction de démarrage, l'équipement bascule en mode opérationnel. Il est alors possible d'effectuer des échanges PDO, mais la communication SDO reste également autorisée.

Boot-Up CANOpen

Procédure de Boot-up

La configuration minimale requise pour les équipements inclut une procédure de démarrage rapide. Cette procédure est illustrée dans le schéma suivant :



Légende

Numéro	Description
1	Mise sous tension du module
2	Après l'initialisation, le module passe automatiquement en mode PRE-OPERATIONAL.
3	Indication du service NMT : START REMOTE NODE
4	Indication du service NMT : PRE-OPERATIONAL
5	Indication du service NMT : STOP REMOTE NODE
6	Indication du service NMT : RESET NODE
7	Indication du service NMT : RESET COMMUNICATION

Activation des objets CANOpen en fonction de la machine d'état

Les croix affichées dans le tableau ci-dessous indiquent les objets CANopen actifs pour chaque état.

	Initialization	Pre-operational	Operational	Stopped
Objet PDO			X	
Objet SDO		X	X	
Emergency		X	X	
Boot-Up	X		X	
NMT		X	X	X

Reset Application

L'équipement passe en mode "Reset Application" :

- après avoir démarré ;
- suite à l'utilisation de la fonction "Reset Node" de la gestion réseau NMT.

Dans cet état, le profil de l'équipement est initialisé et toutes les informations relatives au profil sont réinitialisées avec les valeurs par défaut. Une fois l'initialisation terminée, l'équipement passe à l'état "Reset Communication".

Reset Communication

L'équipement passe en mode "Reset Communication" :

- après avoir été en mode Reset Application ;
- suite à l'utilisation de la fonction "Reset Communication" de la gestion réseau NMT.

Dans cet état, tous les paramètres (valeurs standard, selon la configuration de l'équipement) des objets de communication pris en charge relatifs à l'identification de l'équipement (type, heartbeat, etc. : 1000H - 1FFFH) sont enregistrés dans le répertoire d'objets. L'équipement passe ensuite automatiquement en mode "Init".

Init

L'équipement passe en mode "Init" après avoir été en mode "Reset Communication".

Ce mode permet d'effectuer les opérations suivantes :

- définir les objets de communication requis (SDO, PDO, Emergency) ;
- installer les services CAL (couche application basée sur le réseau) correspondants ,
- configurer la valeur CAN-Controller.

Une fois l'initialisation terminée, l'équipement passe automatiquement en mode "Pre-Operational".

Note : Le module maître CANopen TWDNCO1M et le maître de bus CANopen intégré de Twido Extreme ne prennent pas en charge le mode SYNC.

Pre-Operational

L'équipement passe en mode "Pre-Operational" :

- après avoir été en mode "Init" ou
- à la réception de l'indication NMT "Enter Pre-Operational" s'il était en mode Operational.

Dans ce mode, il est possible de modifier la configuration de l'équipement. Cependant, vous pouvez uniquement utiliser les objets SDO pour lire ou écrire des données relatives aux équipements.

Une fois la configuration terminée, l'équipement passe à l'un des modes suivants, selon l'indication reçue :

- "Stopped" à la réception de l'indication NMT "STOP REMOTE NODE",
- "Operational" à la réception de l'indication NMT "START REMOTE NODE".

Stopped

L'équipement passe en mode "Stopped", après réception de l'indication "Node stop" du service NMT, s'il était en mode "Pre-Operational" ou "Operational".

Dans cet état, il est impossible de modifier la configuration de l'équipement. Aucun service ne permet de lire ou d'écrire des données relatives aux équipements (SDO). Seule la fonction de surveillance de l'esclave ("Node guarding") reste active.

Operational

L'équipement passe en mode "Operational" s'il était en mode "Pre-Operational" à la réception de l'indication "Start Remote Node".

Lors du démarrage du réseau CANopen à l'aide du service NMT "Node start", en mode "Operational", la totalité des fonctionnalités de l'équipement est disponible. Les communications peuvent utiliser les objets PDO et SDO.

Note : Des effets inattendus peuvent se produire suite à la modification de la configuration lorsque l'équipement est en mode "Operational". Par conséquent, les modifications doivent être apportées uniquement en mode "Pre-Operational".

Transmission de Process Data Object (PDO)

Définition d'un objet PDO

Les objets PDO fournissent une interface de communication avec les données de traitement, permettant le transfert de ces données en temps réel. Un ensemble d'objets PDO d'un équipement CANopen décrit les échanges implicites entre l'équipement et ses partenaires de communication sur le réseau.

Ces échanges sont autorisés lorsque l'équipement est en mode "Operational".

Types d'objets PDO

Il existe deux types d'objets PDO :

- les objets PDO transmis par l'équipement (souvent appelés : Transmit PDO, Tx-PDO ou TPDO) ;
- les objets PDO reçus par l'équipement (souvent appelés : Receive PDO, Rx-PDO ou RPDO).

Producers and Consumers de PDO

Les objets PDO s'appuient sur le modèle "Producer / Consumer". L'équipement qui envoie le message PDO est appelé "producer", tandis que celui qui le reçoit est appelé "consumer".

Par conséquent, l'écriture d'une sortie dans le module maître TWDNCO1M ou dans le maître de bus CANopen intégré de Twido Extreme génère l'envoi d'un TPDO associé au maître, contenant la valeur de la sortie à mettre à jour. Dans ce cas, le maître correspond au "producer" du PDO et l'équipement esclave au "consumer" du PDO.

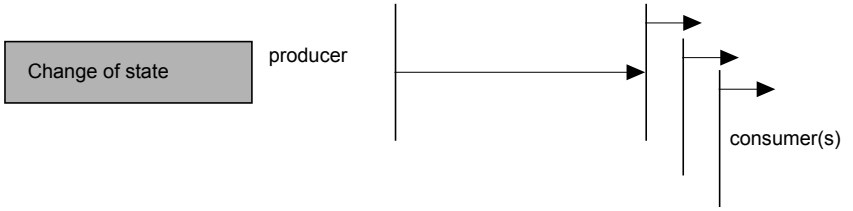
Par opposition, une entrée est mise à jour par le biais de la transmission d'un objet RPDO par le module maître, qui correspond alors au "consumer".

Mode de transmission d'un objet PDO

En plus de définir les données à transporter, il est également possible de configurer le type d'échange de chaque objet PDO.

Les objets PDO peuvent être échangés par le module maître TWDNCO1M ou par le maître de bus CANopen intégré de Twido Extreme avec le mode de transmission suivant :

Numéro de mode	Type de mode	Nom de mode
254 ou 255	Asynchrone	Change of state



Change of state (Modes 254 et 255)

"Change of state" correspond à la modification de la valeur d'une entrée (contrôle d'événement). Les données sont transmises au bus dès qu'une modification est apportée. La fonction de contrôle d'événement optimise l'utilisation de la bande passante du bus, car seules les données modifiées sont transmises, et non la totalité de l'image de process. Il est alors possible d'obtenir un temps de réponse relativement court (lorsque la valeur d'une entrée est modifiée, il n'est pas nécessaire d'attendre la prochaine requête de communication du maître).

Lors de la sélection d'une transmission PDO de type "change of state", vous devez être conscient que plusieurs événements risquent de se produire simultanément, retardant ainsi la transmission des PDO à plus faible priorité vers le bus. Evitez les situations dans lesquelles la modification continuelle d'une entrée avec un PDO de priorité élevée risque de bloquer le bus ("babbling idiot" en anglais).

Note : Il est conseillé de choisir la transmission PDO avec modules d'entrée analogique uniquement si le mode Delta (objet 6426H) ou le temps d'inhibition (objets 1800H à 1804H, sous-index 3) sont configurés de manière à éviter les surcharges de bus.

Accès aux données à l'aide d'échanges explicites (SDO)

Qu'est-ce qu'un objet SDO ?

Les objets Service Data Objects (SDO) permettent d'accéder aux données d'un équipement à l'aide de requêtes explicites.

Le service SDO est disponible lorsque l'état de l'équipement est "Operational" ou "Pre-Operational".

Types d'objets SDO

Il existe deux types d'objets SDO :

- les objets pouvant être lus (objets de type "Download SDO") ;
- les objets pouvant être écrits (objets de type "Upload SDO").

Modèle client/ serveur

Le protocole SDO repose sur un modèle "client/serveur".

Pour un objet de type "Download SDO"

Le client transmet une requête indiquant l'objet à lire.

Le serveur renvoie les données contenues dans cet objet.

Pour un objet de type "Upload SDO"

Le client envoie une requête indiquant l'objet à écrire et la valeur souhaitée.

Une fois l'objet mis à jour, le serveur renvoie un message de confirmation.

Pour un objet SDO non traité

Dans un cas comme dans l'autre, si un objet SDO n'a pas pu être traité, le serveur renvoie un code d'erreur (abort code).

"Node Guarding" et "Life Guarding"

Définition de la notion Life-Time

Le paramètre "Life time" est calculé comme suit :

Life Time = Guard Time x Life Time Factor

L'objet 100CH comprend le paramètre "Guard Time", exprimé en millisecondes. L'objet 100DH comprend le paramètre "Life Time Factor".

Activation de la surveillance

Si l'un de ces deux paramètres est défini sur 0 (valeur par défaut), le module de surveillance n'est pas utilisé (la fonction "Life Guarding" est désactivée).

Pour l'activer, vous devez au moins indiquer la valeur 1 dans l'objet 100DH et spécifier une durée en millisecondes dans l'objet 100CH.

Garantie d'opérations fiables

Pour garantir la fiabilité des opérations, il est conseillé de définir le paramètre "Life time factor" sur 2.

Dans le cas contraire, lorsque le module maître subit un retard (par exemple en raison du traitement des messages dont le niveau de priorité est le plus élevé ou d'un traitement interne lié au "Node Guarding"), son état devient "Pre-Operational" sans qu'aucune erreur ne soit générée.

Importance de la surveillance

Ces deux mécanismes de surveillance sont particulièrement importants pour le système CANopen, étant donné que les équipements fonctionnent généralement dans un mode avec contrôle des événements.

Surveillance de l'esclave

La surveillance est effectuée de la façon suivante :

Phase	Description
1	Le maître règle le paramètre "Remote Frames" (remote transmit requests) sur "Guarding COB-IDs" pour les esclaves à surveiller.
2	Les esclaves concernés répondent en envoyant le message "Guarding". Celui-ci contient le "Status Code" de l'esclave et le "Toggle Bit" dont la valeur doit changer entre deux réponses consécutives.
3	Le maître compare les informations "Status" et "Toggle Bit". Si elles ne correspondent pas à ce que le maître NMT attendait ou si aucune réponse n'est reçue, le maître considère qu'une erreur s'est produite au niveau de l'esclave.

Surveillance du maître

Si des messages de type "Guarding" sont uniquement requis de façon cyclique, l'esclave peut détecter un dysfonctionnement du maître.

Si l'esclave ne reçoit pas de requête du maître dans l'intervalle "Life Time" imparti (erreur de type "Guarding"), il considère qu'un dysfonctionnement du maître s'est produit (mécanisme de surveillance "Watchdog").

Dans ce cas, les sorties correspondantes passent à l'état d'erreur et l'esclave revient en mode "Pre-Operational".

Note : La requête Remote du maître obtient une réponse, même si aucune valeur n'a été saisie pour les paramètres Guard Time et Life Time Factor. La surveillance de la durée n'est activée que lorsque les valeurs de ces deux paramètres sont supérieures à 0. En général, les valeurs du paramètre Guard Time sont comprises entre 250 millisecondes et 2 secondes.

Protocole "Guarding"

La valeur du paramètre" Toggle Bit (t) envoyée dans le premier message "Guarding"est "0".

Ensuite, le bit change ("toggles") dans les messages de surveillance "Guarding" suivants, ce qui permet de savoir si un message a été perdu.

L'en-tête du bus indique l'état du réseau (s) dans les sept bits restants :

Etat du réseau	Réponse
Stopped	0x04 ou 0x84
Pre-operational	0x7F ou 0xFF
Operational	0x05 ou 0x85

Gestion du bus interne

Basculement du bus interne vers l'état "Stop"

Le bus interne passe automatiquement de l'état ''Stop'' à l'état ''Run'' lorsque le module de communication passe de l'état "Pre-operational" à "Operational".

Lorsque le bus interne passe à l'état "Stop", toutes les sorties du module d'expansion* sont définies sur zéro.

L'état des sorties du module de communication reste inchangé.

Configuration des modules d'expansion

Le bus interne est utilisé pour mettre à jour la configuration des paramètres des modules d'expansion* TOR et analogiques.

Les paramètres sont envoyés au module de communication lorsque l'état du bus est ''Stop''.

Ces nouveaux paramètres de configuration sont reconnus lorsque le bus passe à l'état ''Run''.

Note : * L'automate Twido Extreme TWDLEDCK1 ne prend pas en charge les modules d'expansion.

10.2 Mise en œuvre du bus CANopen

Vue d'ensemble

Introduction

Cette section décrit les procédures de mise en œuvre du bus de terrain CANopen sur le système de l'automate Twido, en utilisant le module maître CANopen TWDNCO1M ou le bus CANopen intégré de Twido Extreme.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble	267
Configuration matérielle	269
Méthode de configuration	270
Déclaration d'un module maître CANopen	272
Configurateur CANopen	273
Déclaration des équipements esclaves CANopen sur le réseau	275
Mappage des objets CANopen	279
Liaison des objets CANopen	283
Symbolisation des objets CANopen	285
Adressage des PDO du module maître CANopen	287
Programmation et diagnostic du bus de terrain CANopen	289
Echange sous tension CANopen pour Twido Extreme	296

Vue d'ensemble

Configuration matérielle et logicielle requise

Les solutions matérielles et logicielles ci-dessous sont requises pour la mise en œuvre d'un bus CANopen sur la base compacte ou modulaire Twido :

Matériel	Configuration requise
Automate Twido (base compacte ou modulaire)	**Base compacte :** • TWDLC•24DRF • TWDLC••40DRF **Base modulaire :** • TWDLMDA20••• • TWDLMDA40•••
Module maître CANopen	1 module maître CANopen : TWDNCO1M
Equipements esclaves CANopen	16 esclaves CANopen maximum
Connecteurs et câbles CANopen	
Câble de programmation de l'automate Twido	• Câble série : TSX PCX1031 • Câbles USB : TSX CUSB485 et TSX CRJMD25 • Câble de communication avec prise Mini-DIN à une extrémité, l'autre extrémité étant libre : TSX CX100

Logiciel	Configuration requise
Logiciel de configuration de l'automate Twido	TwidoSuite

Les solutions matérielles et logicielles ci-dessous sont requises pour la mise en œuvre d'un bus CANopen sur votre automate Twido Extreme :

Matériel	Configuration requise
Base automate Twido Extreme	TWDLEDCK1
Equipements esclaves CANopen	16 esclaves CANopen maximum
Connecteurs et câbles CANopen	Reportez-vous au Guide de référence du matériel de Twido Extreme.
Câble de programmation pour Twido Extreme	• Câble série : VW3 A8106 • Câbles USB : TSX CUSB485 et TWD XCAFJ010 • Adaptateur Bluetooth pour l'automate : VW3 A8114

Logiciel	Configuration requise
Logiciel de configuration de l'automate Twido	TwidoSuite

Procédure de mise en œuvre du réseau CANopen

La procédure ci-dessous est destinée à vous guider lors de l'installation, de la configuration et de l'utilisation du réseau CANopen :

Etape	Description
1	Configuration matérielle
2	Méthode de configuration
3	Déclaration du module maître CANopen
4	Déclaration des équipements esclaves CANopen sur le réseau
5	Mappage des objets CANopen
6	Liaison des objets CANopen
7	Symbolisation des objets CANopen
8	Diagnostics du réseau CANopen
Les sous-sections ci-après fournissent une description détaillée de chacune des étapes de la procédure.	

Configuration matérielle

Câblage CANopen de Twido Extreme

Pour la base automate Twido Extreme, reportez-vous à la rubrique Network Wiring, TwidoSuite Programming Software, TwidoExtreme Hardware Guide.

Installation du module maître TWDNCO 1M

Pour les autres bases automates compactes ou modulaires, installez le module maître TWDNCO1M sur un automate Twido (montage sur panneau ou sur rail DIN) et connectez-le au bus interne de l'automate (la méthode de raccordement est la même que celle qui est décrite dans la section Assemblage d'un module d'expansion d'E/S (Assemblage d'un module d'expansion d'E/S à une base, Guide de référence du matériel, Modules d'E/S TOR)). Procédez comme suit :

Etape	Action	Description
1	Préparation de l'installation	Consultez le *Guide de référence du matériel d'automates programmables Twido (TWD USE 10AE)* pour plus de détails sur : ● les positions de montage correctes des modules Twido ; ● l'ajout de composants Twido sur un rail DIN, ou leur suppression ; ● le montage direct sur un panneau ; ● les dégagements minimaux des modules dans un panneau de commande.
2	Montage du module TWDNCO1M	Installez le module maître TWDNCO1M sur un panneau ou sur un rail DIN. Pour plus d'informations, reportez-vous à la section TwdoHW - Installation d'un module d'expansion.
3	Connexion du module au bus de l'automate Twido	Connectez le module maître CANopen au bus interne de l'automate Twido (pour plus d'informations, reportez-vous à la rubrique TwdoHW - Installation d'un module d'expansion).
4	Câblage et raccordement CANopen	Pour connecter l'alimentation électrique du bus CAN au circuit de transmission, suivez les instructions relatives au câblage et au raccordement contenues sous la rubrique Câblage et raccordement CANopen.

Méthode de configuration

Vue d'ensemble

La configuration CANopen est effectuée par l'intermédiaire de l'outil CANopen Configuration Tool disponible sous TwidoSuite.

Note :
1. La configuration de l'esclave, du maître et du réseau CANopen ainsi que celle des paramètres de communication sont effectuées en mode local uniquement.
2. Aucune modification de la configuration CANopen n'est autorisée en mode connecté.
3. Seuls certains paramètres peuvent être ajustés en mode connecté, tels que les paramètres d'adressage des objets PDO %IWC et %QWC.

Méthode de configuration

Le tableau ci-dessous présente les différentes phases de mise en œuvre logicielle du bus CANopen :

Mode	Phase	Description
Local	Déclaration du maître CANopen	Dans la fenêtre Description, ajoutez un module maître CANopen TWDNCO1M à votre application Twido.[1]
	[1]Cela n'est pas nécessaire dans le cas de l'automate Twido Extreme TWDLDCK1, car le bus maître CANopen est intégré à la base automate.	
	Configuration du réseau CANopen	Configurez le réseau CANopen comme suit : ● importez les fichiers EDS de tous les équipements esclaves dans le catalogue du réseau ; ● ajoutez les équipements esclaves du catalogue dans le réseau CANopen.
	Mappage des PDO	Effectuez le mappage des objets TPDO et RPDO de chaque équipement esclave déclaré sur le réseau.
	Liaison des PDO	Associez chaque objet PDO esclave au PDO du module maître correspondant.
Local ou connecté	Symbolisation (optionnel)	Symbolisation des variables associées aux équipements esclaves.
	Programmation	Programmation de la fonction CANopen.
Connecté	Transfert	Transfert de l'application dans l'automate.
	Mise au point	Mise au point de l'application à l'aide : ● de l'écran de mise au point permettant d'une part la visualisation des esclaves (adresse, paramètres), et d'autre part l'adressage des esclaves aux adresses souhaitées. ● des écrans de diagnostic permettant d'identifier les défauts.

Note : La déclaration et la suppression du module maître CANopen TWDNCO1M sur le bus d'expansion se déroulent comme pour un autre module d'expansion. Toutefois, seul un module maître CANopen peut être installé sur le bus d'expansion Twido. Le programme TwidoSuite n'accepte aucun module CANopen supplémentaire.

Précautions avant la connexion

Avant de connecter (de façon logicielle) le PC à l'automate et pour éviter tout problème de détection :

- Assurez-vous qu'aucun esclave n'est installé sur le bus à l'adresse 127 (il s'agit d'une adresse réservée et prédéfinie qui est attribuée au module maître TWDNCO1M ou au maître de bus CANopen intégré de Twido Extreme).
- Assurez-vous qu'aucun esclave n'est installé sur le bus CANopen avec des adresses en double.

Déclaration d'un module maître CANopen

Vue d'ensemble

Cette section décrit comment déclarer un module maître de bus CANopen sur le bus interne de votre automate compact ou modulaire.

Note : Les informations de cette section ne s'appliquent pas à la base automate Twido Extreme TWDLDCK1 qui intègre un module maître de bus CANopen.

Marche à suivre

Le tableau ci-dessous présente les différentes étapes nécessaires à la déclaration du module maître CANopen.

Etape	Action	Commentaire
1	Sélectionnez l'étape **Description** de l'interface TwidoSuite.	Reportez-vous à la section (Fenêtre Décrire l'application, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
2	Affichez le catalogue des produits et sélectionnez un module **TWDNCO1M** à ajouter à la description du système.	Reportez-vous à la section (Positionnement d'un module, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). **Remarque :** Vous pouvez insérer un module maître TWDNCO1M dans n'importe quel emplacement d'expansion disponible. Il est ensuite possible de le déplacer à gauche ou à droite des autres modules d'expansion. Vous pouvez continuer à ajouter d'autres modules d'expansion (jusqu'à 7, module TWDNCO1M inclus). **Remarque :** Vous ne pouvez utiliser qu'un seul module maître CANopen TWDNCO1M. Seuls les automates TWDC•A24DRF, TWDC••40DRF, TWDLMDA20••• et TWDLMDA40••• sont pris en charge.

Configurateur CANopen

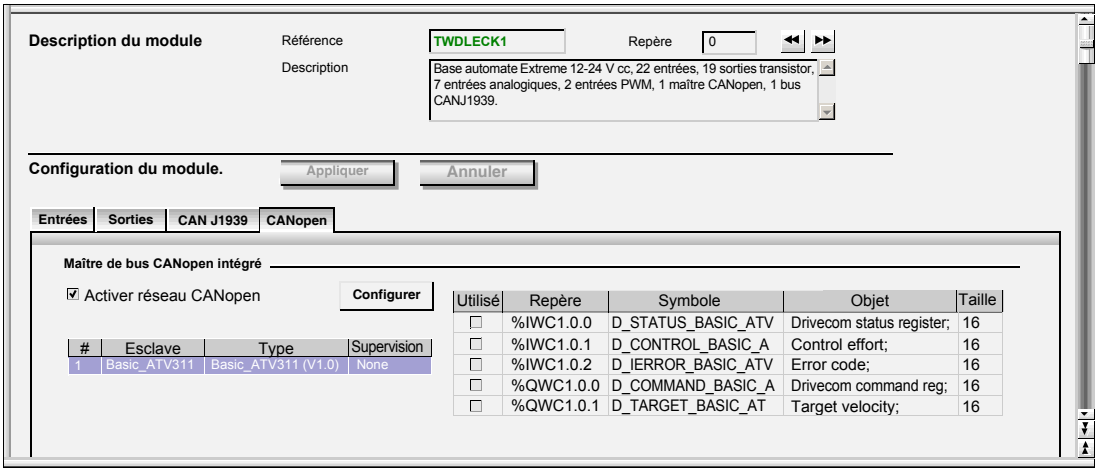
Vue d'ensemble

Cette section explique comment accéder à l'outil CANopen configuration tool (voir *p. 275*)

Configurateur CANopen

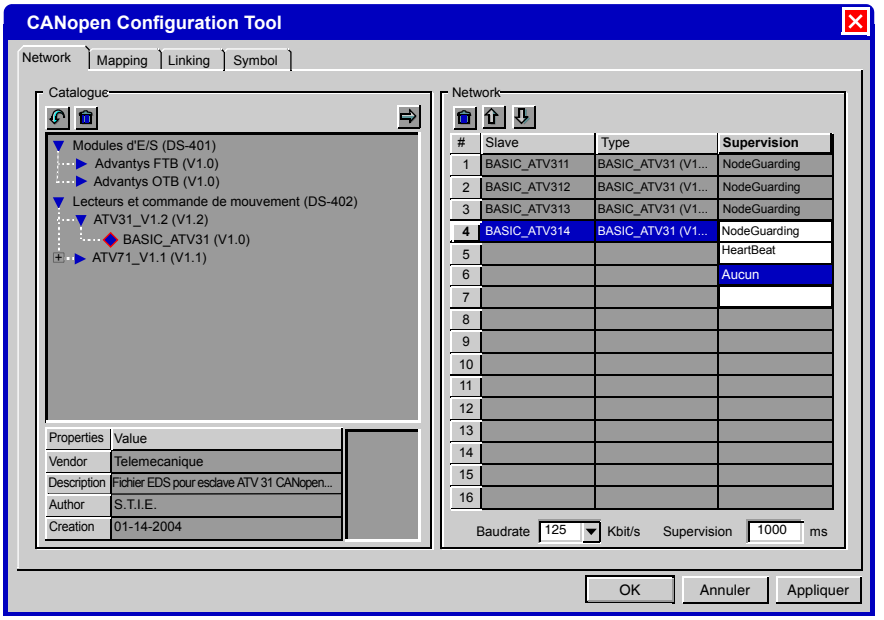
Dans l'interface utilisateur TwidoSuite, sélectionnez la tâche **Programme** → **Configurer** → **Configurer le matériel**.

Dans le volet graphique (Volet graphique de la configuration matérielle, Aide en ligne TwidoSuite) :



- Pour TWDLEDCK1 Twido Extreme, sélectionnez l'onglet CANopen et cliquez sur **Configurer**.
- Pour les autres bases automate modulaires et compactes, sélectionnez le module **TWDNCO1M** dans le volet graphique supérieur, puis cliquez sur Configurer dans le volet de configuration inférieur (pour les automates compacts et modulaires, il n'y a pas d'onglet).

Résultat : l'outil **CANopen Configuration Tool** apparaît à l'écran dans le volet Configuration (Volet de configuration matérielle,Aide en ligne TwidoSuite), comme illustré ci-dessous :



Note : Vous pouvez désactiver le maître de bus CANopen intégré de Twido Extreme en décochant la case **Activer réseau CANopen** de l'onglet CANopen du volet de configuration. Cette opération permet de libérer de la mémoire dédiée à CANopen. Désactiver le maître de bus CANopen intégré de Twido Extreme efface la configuration réseau CANopen.

Déclaration des équipements esclaves CANopen sur le réseau

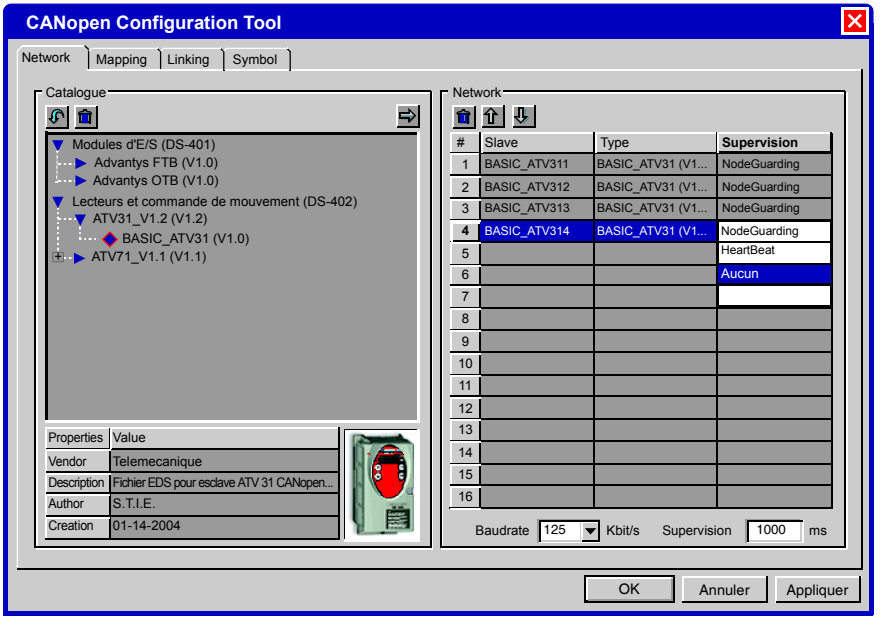
Vue d'ensemble

La déclaration des équipements esclaves CANopen sur le réseau est un processus en trois étapes qui consiste à effectuer les opérations suivantes :

1. importer les fichiers EDS des équipements esclaves CANopen dans le catalogue du configurateur CANopen Twido ;
2. développer le réseau CANopen en lui ajoutant jusqu'à 16 équipements esclaves du catalogue ;
3. configurer les paramètres de gestion du réseau (vitesse du réseau et protocole de gestion des erreurs).

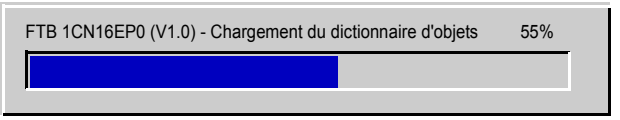
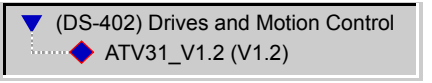
Boîte de dialogue Réseau

La déclaration des équipements esclaves et la configuration du réseau CANopen sont réalisées par l'intermédiaire de la boîte de dialogue **Réseau** du configurateur CANopen de TwidoSuite, comme indiqué ci-dessous :



Importation des profils des équipements esclaves

Le tableau ci-dessous indique comment importer les profils (fichiers .EDS) des équipements esclaves CANopen dans le catalogue de l'outil CANopen Configuration Tool :

Etape	Action
1	Dans la section **Catalogue** de la boîte de dialogue Réseau, cliquez sur l'icône **Importer** . **Résultat :** la boîte de dialogue Ouvrir du système d'exploitation s'affiche.
2	Recherchez le dossier contenant les fichiers **EDS** des équipements esclaves CANopen à ajouter au catalogue. **Résultat :** le nom des fichiers EDS disponibles apparaît dans la boîte de dialogue Ouvrir :
3	Sélectionnez un fichier EDS **("nomfichier".EDS)** dans la liste et cliquez sur **Ouvrir**. **Résultat :** l'outil CANopen Configuration Tool charge le dictionnaire des objets de l'équipement sélectionné. **Remarque :** Selon la taille du fichier EDS sélectionné, ce processus peut prendre plusieurs minutes. Une barre de progression indique l'avancement du chargement, comme illustré dans l'exemple ci-dessous : 
4	Une fois le chargement terminé, répétez la procédure des étapes 2 et 3 pour chacun des profils d'équipement esclave que vous souhaitez ajouter au catalogue. **Remarque :** Il n'est nécessaire de réaliser cette procédure qu'une seule fois, car tous les profils d'équipement et dictionnaires d'objets répertoriés dans le catalogue chargé sont stockés par TwidoSuite.
5	Pour afficher les propriétés d'un équipement esclave CANopen : **1.** Cliquez deux fois sur le type d'équipement répertorié dans le catalogue. **Exemple :** . **Résultat :**  . **2.** Cliquez sur le profil de l'équipement esclave (par exemple, ATV31 V1.2). **Résultat :** Les propriétés de l'équipement esclave CANopen sélectionné s'affichent dans la partie inférieure de la section Catalogue. Elles fournissent les informations suivantes : • le nom du fournisseur (par exemple, Telemecanique) ; • le profil de l'équipement esclave (par exemple, fichier EDS de description pour esclave CANopen ATV 31) ; • le nom de l'auteur (par exemple, Auteur S.T.I.E.) ; • la date de création du profil (par exemple, 01-14-2004).

Etape	Action
6	Pour supprimer le profil d'un équipement esclave du catalogue, sélectionnez le nom de l'équipement dans la fenêtre Catalogue et cliquez sur l'icône **Supprimer** . **Remarque :** Vous pouvez stocker dans le catalogue CANopen du réseau plus de profils que nécessaire pour la configuration de votre bus CANopen. Les profils déjà chargés dans le catalogue peuvent être configurés pour une utilisation ultérieure.
7	Cliquez sur le bouton **Appliquer** pour valider les modifications apportées au catalogue et enregistrer les profils d'esclave dans le projet TwidoSuite.

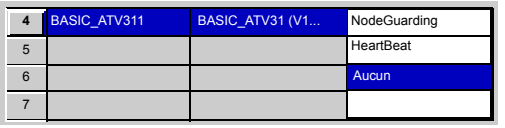
Développement du réseau CANopen

Le tableau ci-dessous explique comment déclarer les équipements esclaves sur un réseau CANopen Twido (notez que vous ne pouvez déclarer que les équipements dont les profils EDS figurent déjà dans le catalogue ou que vous avez vous-même déjà ajoutés au catalogue).

Etape	Action
1	Dans la section **Catalogue** de la boîte de dialogue Réseau, sélectionnez le profil de l'équipement esclave dans la liste des équipements disponibles déjà enregistrés dans le catalogue. **Résultat :** l'icône **Ajouter** apparaît dans le coin supérieur droit du cadre du catalogue.
2	Cliquez sur l'icône **Ajouter** . **Résultat :** l'équipement esclave est ajouté à la table des équipements esclaves du réseau. **Remarques :** • Vous pouvez déclarer jusqu'à 16 équipements esclaves sur un réseau CANopen Twido. • L'équipement esclave nouvellement déclaré prend l'adresse de nœud dotée du plus petit index disponible (par exemple, si des équipements esclaves sont déclarés aux adresses de nœud 1, 2 et 4, l'équipement que vous ajouterez prendra l'adresse 3 par défaut).
3	Vous pouvez attribuer n'importe quelle adresse de nœud disponible (entre 1 et 16) à l'équipement esclave. Pour déplacer un équipement vers l'adresse souhaitée, utilisez les flèches vers le **haut/bas** / .
4	Répétez les étapes 1 à 3 pour chaque équipement esclave à déclarer sur le réseau CANopen.
5	Pour supprimer un équipement esclave du réseau, sélectionnez son nom dans la table des esclaves et cliquez sur l'icône **Supprimer** .
6	Cliquez sur le bouton **Appliquer** pour valider les modifications et enregistrer la configuration du réseau dans le projet TwidoSuite.

Configuration des paramètres de gestion du réseau

La procédure ci-dessous explique comment configurer les paramètres de gestion du réseau, notamment les paramètres Baudrate (vitesse du réseau), life-time (durée de vie) et le protocole de gestion des erreurs.

Etape	Action
1	Dans la boîte de dialogue Réseau, sélectionnez le paramètre Baudrate (vitesse du réseau) dans la liste déroulante : **125, 250 (valeur par défaut), 500.** **Remarque :** Veillez à configurer le paramètre Baudrate de chacun des équipements esclaves déclarés sur le réseau de sorte qu'il corresponde exactement à la vitesse du réseau définie ici. Sans cette opération, les communications du réseau CANopen risqueraient de ne pas fonctionner correctement.
2	Configurez la valeur Life-time. Ce paramètre définit le temps de cycle des communications qui sera implémenté dans le champ de supervision de chaque équipement esclave, comme expliqué plus bas à l'étape 3. **Remarque :** La valeur 0 n'est pas autorisée dans ce champ.
3	Cliquez dans le champ **Supervision** pour configurer les options relatives au protocole de gestion des erreurs des équipements esclaves déclarés dans la table des esclaves du réseau. **Résultat :** les options de supervision disponibles prises en charge par l'équipement sélectionné apparaissent dans une liste, comme indiqué ci-dessous : 
4	Choisissez le protocole de gestion des erreurs à utiliser pour gérer les communications entre le module maître TWDNCO1M ou le maître de bus CANopen intégré de Twido Extreme et l'équipement esclave sélectionné : ● NodeGuarding ● Heartbeat ● None
5	Si l'option de supervision est définie sur **Aucun** dans la table des esclaves du réseau, les sorties ne sont pas rétablies à leurs valeurs de repli dans le cas d'une interruption de la connexion (*) entre cet esclave et le module maître TWDNCO1M ou le maître de bus CANopen intégré de Twido Extreme. (*) cette déconnexion peut avoir pour origine : ● une déconnexion du câble du bus d'expansion reliant le module maître CANopen TWDNCO1M à la base automate Twido (ne s'applique pas à l'automate Twido Extreme) ; ● une déconnexion de cet esclave CANopen du bus CANopen Twido ; ● un câble de bus défectueux ; ● une commande de remise à zéro de TwidoSuite (Mode connecté → Microprogramme / Remise à zéro) ; ● une commande de configuration de téléchargement TwidoSuite (Mode connecté → Télécharger) ; ● une commande de téléchargement de microprogramme sur le module maître TWDNCO1M (ou la base automate Twido Extreme) via TwidoSuite (Mode connecté → Télécharger microprogramme) (ne s'applique pas à l'automate Twido Extreme).
6	Cliquez sur le bouton **Appliquer** pour valider les modifications et enregistrer la configuration du réseau dans le projet TwidoSuite.

Mappage des objets CANopen

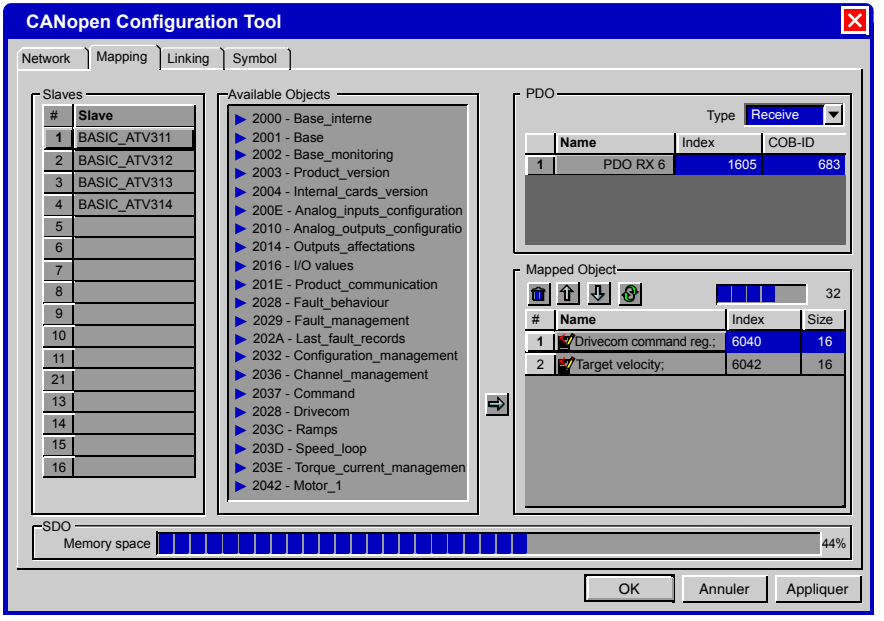
Vue d'ensemble

La boîte de dialogue **Mapping** de l'outil CANopen Configuration Tool vous permet de configurer les objets PDO de chaque équipement esclave déclaré sur le réseau.

Boîte de dialogue Mapping

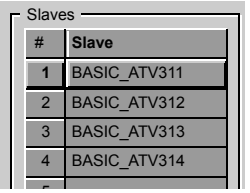
Cliquez sur l'onglet **Mapping**.

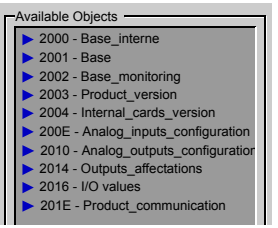
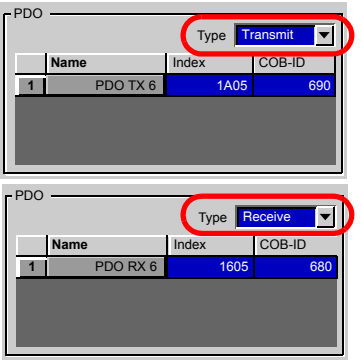
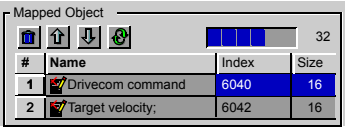
Résultat : L'outil CANopen Configuration Tool apparaît à l'écran, comme illustré ci-dessous :



Mappage des objets

Pour savoir comment configurer les objets TPDO et RPDO de chacun des équipements esclaves à l'aide de la boîte de dialogue Mapping, suivez les instructions ci-dessous :

Etape	Action
1	Dans le cadre **Slaves**, cliquez sur le nom de l'équipement pour sélectionner l'esclave pour lequel vous voulez configurer les objets PDO.
2	**Exemple :** Famille DS-402 Drives and Motion Control appelée BASIC_ATV311. Notez que les noms d'esclave et les adresses de nœud apparaissent tels qu'ils ont été définis à l'étape précédente de configuration du réseau (reportez-vous à la *p. 275*). 

Etape	Action
3	**Résultat :** **1.** Les objets CANopen pris en charge par l'esclave sélectionné apparaissent dans le cadre **Available Objects**, comme indiqué dans l'exemple ci-dessous :  **2.** La fenêtre **PDO** indique les Transmit-PDOs (PDO TX) prédéfinis par défaut pour l'esclave sélectionné. En outre, vous pouvez utiliser la liste **Type** pour afficher ou non les Receive-PDOs (PDO RX). Dans cet exemple, la famille BASIC_ATV311 DS-402 Drives and Motion Control prend en charge un Transmit-PDO (PDO TX) et un Receive-PDO (PDO RX), comme indiqué ci-dessous :  **3.** Le mappage prédéfini de chacun des PDO sélectionnés est également affiché dans le cadre **Mapped Objects**. : 
4	Il vous est possible de personnaliser le mappage des objets PDO dans le cadre **Mapped Objects**. Un objet RPDO ou TPDO constitue un objet de 64 octets pouvant comporter jusqu'à huit objets mots de 8 octets ou quatre objets mots de 16 octets chacun ou toute autre combinaison de ces deux types d'objets mots à condition que la limite totale autorisée (64 octets) ne soit pas dépassée. Pour personnaliser le mappage d'un objet PDO, reprenez à l'étape 5 et poursuivez jusqu'à la dernière étape en tenant compte de ces considérations.

Etape	Action
5	Sélectionnez l'objet PDO de l'esclave voulu (voir étape 2) pour lequel vous souhaitez modifier le mappage à partir du cadre **PDO**. **Exemple :** Sélectionnez le premier Transmit-PDO (PDO TX 6). **Résultat :** Le mappage prédéfini de l'objet PDO (ou le mappage personnalisé actif) apparaît dans le cadre Mapped Objects.
6	Pour supprimer un objet mot inutilisé de la structure de mappage des objets PDO, sélectionnez-le (objet indexé de 1 à 8) et cliquez sur l'icône de **suppression** .
7	Dans le cadre **Available Objects**, sélectionnez l'objet mot dans la famille d'objets à mapper, puis cliquez sur l'icône **Add** pour l'ajouter à la structure **Mapped Objects**. **Remarque :** Pour restituer la structure de mappage par défaut de l'objet PDO sélectionné, cliquez sur l'icône **Default** .
8	Pour modifier l'adresse d'un objet mot dans la structure de l'objet PDO mappée, utilisez les flèches vers le **haut/bas** .
9	Cliquez sur le bouton **Apply** pour valider les modifications apportées à la structure mappée de l'objet PDO et enregistrer le projet TwidoSuite.
10	Répétez les étapes 5 à 9 pour chaque mappage d'objet PDO à configurer.
11	**Remarques sur le bilan mémoire :** • **Bilan mémoire du PDO :** Le bilan mémoire du PDO peut être contrôlé par l'intermédiaire de la barre d'état située dans l'angle supérieur droit du cadre Mapped Objects : . • **Bilan mémoire supplémentaire du SDO :** Les objets mots et PDO prédéfinis n'utilisent pas de mémoire SDO supplémentaire. Cependant, les opérations d'ajout et de suppression d'objets mots associées à la structure de mappage PDO requièrent l'utilisation de mémoire système supplémentaire. L'utilisation actuelle de la mémoire SDO est indiquée dans la barre d'état située dans l'angle inférieur de la boîte de dialogue Mapping : .

Liaison des objets CANopen

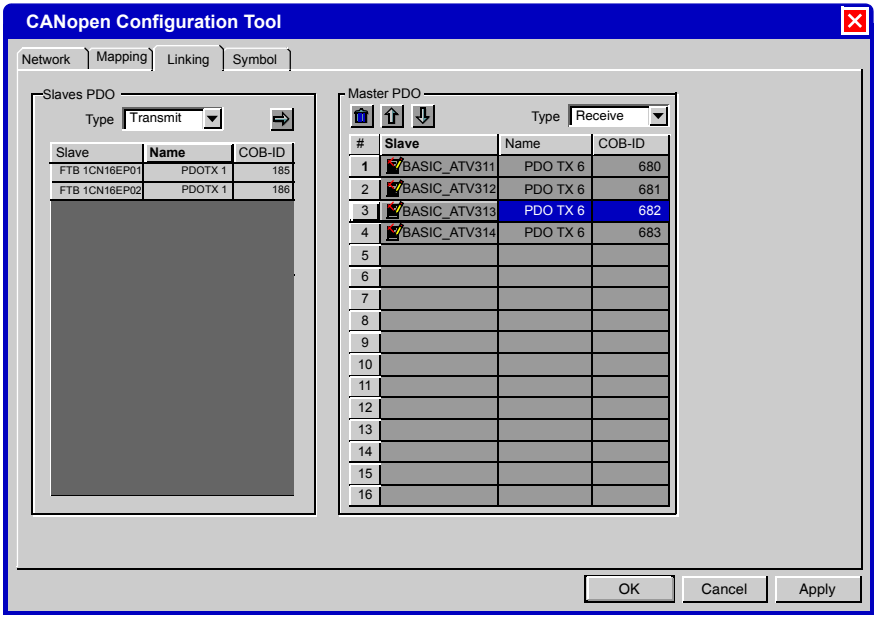
Vue d'ensemble

La boîte de dialogue **Linking** de l'outil CANopen Configuration Tool permet d'établir une liaison physique entre les objets PDO des équipements esclaves sélectionnés et ceux du module maître CANopen TWDNCO1M ou du maître de bus CANopen intégré de Twido Extreme.

Boîte de dialogue Linking

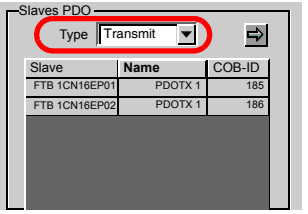
Cliquez sur l'onglet **Linking**.

Résultat : L'outil CANopen Configuration Tool apparaît à l'écran, comme illustré ci-dessous :



Liaison des objets

Pour savoir comment définir une liaison physique entre les objets PDO de l'équipement esclave et ceux du module maître à partir de la boîte de dialogue Linking, suivez les instructions ci-dessous :

Etape	Action
1	Dans le cadre **Slave PDOs**, sélectionnez le **type** de l'objet PDO (Receive ou Transmit). **Résultat :** Tous les objets PDO esclaves appartenant au type sélectionné apparaissent dans le cadre Slave PDOs, comme indiqué ci-dessous :  **Remarque :** Lorsque vous sélectionnez le type Receive ou Transmit dans le cadre Slave PDOs, le cadre Master PDOs affiche automatiquement le type inverse : Transmit ou Receive, respectivement.
2	Dans le cadre **Slave PDOs**, sélectionnez l'objet PDO pour lequel vous souhaitez créer une liaison avec le module maître CANopen TWDNCO1M ou avec le maître de bus CANopen intégré de Twido Extreme, puis cliquez sur l'icône **Add** pour l'ajouter à la liste des liaisons **Master PDOs**. **Remarque :** Le module maître TWDNCO1M et le maître de bus CANopen intégré de Twido Extreme prennent en charge un maximum de 16 liaisons TPDO et 16 liaisons RPDO.
3	Pour modifier l'indexation du repère d'une liaison PDO dans le cadre Master PDOs, utilisez les flèches vers le **haut/bas** .
4	Pour supprimer une liaison PDO inutilisée du cadre Master PDOs, sélectionnez l'objet en question (indexé de 1 à 16) et cliquez sur l'icône de **suppression** .
5	Cliquez sur le bouton **Apply** pour valider les modifications apportées à la structure mappée d'objets PDO et enregistrer le projet TwidoSuite.
6	Répétez les étapes 1 à 5 pour chaque objet PDO esclave à lier au module maître CANopen.

Symbolisation des objets CANopen

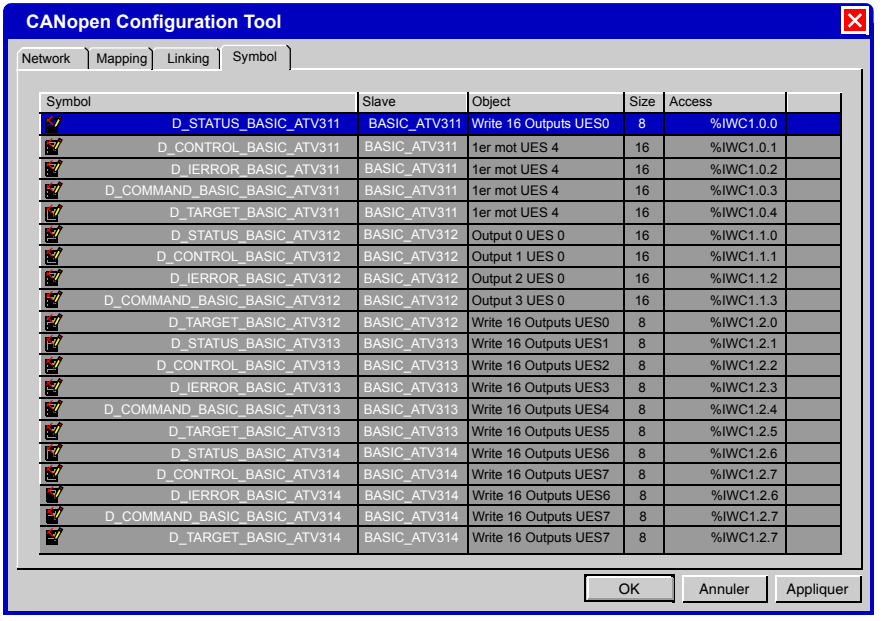
Vue d'ensemble

La boîte de dialogue **Symbole** permet de fixer la symbolisation des variables associées au module maître CANopen.

Boîte de dialogue Symbole

Cliquez sur l'onglet **Symbole**.

Résultat : L'outil CANopen Configuration Tool apparaît à l'écran, comme illustré ci-dessous :



Symbol	Slave	Object	Size	Access
D_STATUS_BASIC_ATV311	BASIC_ATV311	Write 16 Outputs UES0	8	%IWC1.0.0
D_CONTROL_BASIC_ATV311	BASIC_ATV311	1er mot UES 4	16	%IWC1.0.1
D_IERROR_BASIC_ATV311	BASIC_ATV311	1er mot UES 4	16	%IWC1.0.2
D_COMMAND_BASIC_BASIC_ATV311	BASIC_ATV311	1er mot UES 4	16	%IWC1.0.3
D_TARGET_BASIC_ATV311	BASIC_ATV311	1er mot UES 4	16	%IWC1.0.4
D_STATUS_BASIC_ATV312	BASIC_ATV312	Output 0 UES 0	16	%IWC1.1.0
D_CONTROL_BASIC_ATV312	BASIC_ATV312	Output 1 UES 0	16	%IWC1.1.1
D_IERROR_BASIC_ATV312	BASIC_ATV312	Output 2 UES 0	16	%IWC1.1.2
D_COMMAND_BASIC_BASIC_ATV312	BASIC_ATV312	Output 3 UES 0	16	%IWC1.1.3
D_TARGET_BASIC_ATV312	BASIC_ATV312	Write 16 Outputs UES0	8	%IWC1.2.0
D_STATUS_BASIC_ATV313	BASIC_ATV313	Write 16 Outputs UES1	8	%IWC1.2.1
D_CONTROL_BASIC_ATV313	BASIC_ATV313	Write 16 Outputs UES2	8	%IWC1.2.2
D_IERROR_BASIC_ATV313	BASIC_ATV313	Write 16 Outputs UES3	8	%IWC1.2.3
D_COMMAND_BASIC_BASIC_ATV313	BASIC_ATV313	Write 16 Outputs UES4	8	%IWC1.2.4
D_TARGET_BASIC_ATV313	BASIC_ATV313	Write 16 Outputs UES5	8	%IWC1.2.5
D_STATUS_BASIC_ATV314	BASIC_ATV314	Write 16 Outputs UES6	8	%IWC1.2.6
D_CONTROL_BASIC_ATV314	BASIC_ATV314	Write 16 Outputs UES7	8	%IWC1.2.7
D_IERROR_BASIC_ATV314	BASIC_ATV314	Write 16 Outputs UES6	8	%IWC1.2.6
D_COMMAND_BASIC_BASIC_ATV314	BASIC_ATV314	Write 16 Outputs UES7	8	%IWC1.2.7
D_TARGET_BASIC_ATV314	BASIC_ATV314	Write 16 Outputs UES7	8	%IWC1.2.7

Symbolisation des objets

Pour savoir comment définir des symboles pour les variables d'objets CANopen à l'aide de la boîte de dialogue Symbole, suivez les instructions ci-dessous :

Etape	Action
1	A partir du champ **Symbole**, cliquez deux fois sur l'icône d'édition correspondant à la variable à représenter par un symbole. **Résultat :** La zone de texte associée au symbole est activée et le curseur est aligné à droite. **Remarque :** Le symbole signifie que le champ **Symbole** est en lecture seule.
2	Saisissez un nom explicite. Les symboles valides peuvent comporter jusqu'à 32 caractères et doivent contenir uniquement des lettres (A à Z), des chiffres (0 à 9) et des traits de soulignement (_) (les caractères "/", "%", les espaces et les caractères spéciaux ne sont pas autorisés). **Remarque :** Pour plus d'informations sur l'édition des symboles, reportez-vous à la *p. 51*.
3	Cliquez sur le bouton **Appliquer** pour valider les modifications apportées à la table des symboles et enregistrer le projet TwidoSuite.
4	Répétez les étapes 1 à 3 pour chaque variable à représenter par un symbole.

Adressage des PDO du module maître CANopen

Présentation

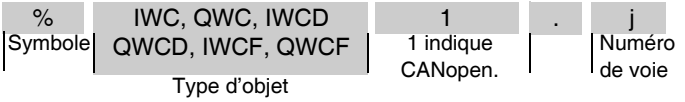
Cette sous-section décrit l'adressage des entrées et sorties des PDO du module maître CANopen.

Pour éviter toute confusion avec les E/S distantes, une nouvelle désignation est utilisée pour la syntaxe des objets CANopen : %I**WC** par exemple.

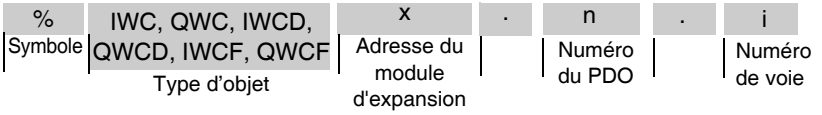
Illustration

Rappel des principes d'adressage :

Pour l'automate Twido Extreme TWDLEDCK1 :



Pour les autres bases automates compactes et modulaires :



Valeurs spécifiques

Le tableau ci-dessous attribue des valeurs spécifiques aux objets des esclaves CANopen :

Elément	Valeurs	Commentaire
IWC	-	Image de l'entrée physique des PDO.
QWC	-	Image de la sortie physique des PDO.
IWCD	-	Même utilisation qu'IWC, mais au format double mot.
QWCD	-	Même utilisation que QWC, mais au format double mot.
IWCF	-	Même utilisation qu'IWC, mais au format en virgule flottante.
QWCF	-	Même utilisation que QWC, mais au format en virgule flottante.
x	1 à 7	Adresse d'un module maître CANopen TWDNCO1M installé sur le bus d'expansion Twido. Pour Twido Extreme x=1
n	0 à 15	Numéro de PDO (selon l'index PDO)
i	0 à 7	Numéro de voie (selon le sous-index PDO)

Exemple

Le tableau suivant présente un exemple d'adressage des PDO :

Objet d'E/S	Description
%IWC4.1.0	PDO numéro 1, entrée 0 du sous-index du module CANopen situé à l'adresse 4 du bus d'expansion Twido.

Echanges implicites

Les objets décrits ci-dessous sont échangés de façon implicite, c'est-à-dire qu'ils sont échangés de façon automatique à chaque cycle de l'automate.

Programmation et diagnostic du bus de terrain CANopen

Echanges explicites

Des objets (mots et bits) associés au bus de terrain CANopen apportent des informations (par exemple : fonctionnement du bus, état des esclaves, etc.) et des commandes supplémentaires pour effectuer une programmation avancée de la fonction CANopen.

Ces objets sont échangés de façon explicite entre l'automate Twido et le module maître CANopen par le bus d'expansion :

- sur demande de l'utilisateur du programme à l'aide de l'instruction CAN_CMD (reportez-vous à la section Présentation de l'instruction CAN_CMD) ;
- via l'écran de mise au point ou la table d'animation.

Mots système spécifiques réservés au module maître CANopen

Les mots système réservés dans l'automate Twido pour le module maître CANopen TWDNCO1M permettent de connaître l'état du réseau : %SW8x (x=1-7) est réservé au module maître CANopen à l'adresse d'expansion x du bus Twido. Seuls les 7 premiers bits de ces mots sont utilisés et sont en lecture seule.

Pour le maître de bus CANopen intégré de Twido Extreme, le mot système spécifique réservé est toujours %SW81 (les mots %SW82 à %SW87 ne sont pas utilisés).

Le tableau suivant présente les bits utilisés :

Mots système	Bit	Description
%SW8x (x=1-7)	0	Etat de la configuration du module maître CANopen (= 1 si configuration OK, sinon 0)
	1	Mode de marche du module maître CANopen (=1 si échange de données activé, sinon 0)
	2	Système en mode d'arrêt (= 1 si le mode local est activé, sinon 0)
	3	Instruction CAN_CMD terminée (= 1 si commande terminée, sinon 0 si commande en cours)
	4	Erreur instruction CAN_CMD (= 1 si erreur dans instruction, sinon 0)
	5	Erreur d'initialisation (= 1)
	6	Perte de message, erreur d'alimentation (= 1)

Exemple d'utilisation (pour le module maître CANopen installé à l'adresse d'expansion 1 du bus Twido) :

Avant d'utiliser une instruction CAN_CMD, le bit %SW81:X3 doit être vérifié pour savoir si une instruction n'est pas en cours : vérifiez que %SW81:X3 = 1.

Pour savoir si l'instruction a ensuite été correctement exécutée, vérifiez que le bit %SW81:X4 est égal à 0.

Mots système spécifiques réservés au module esclave CANopen

Les mots système réservés de %SW20 à %SW27 permettent de connaître l'état actuel des 16 modules esclaves CANopen avec des adresses de nœud comprises entre 1 et 16. Le contenu de ces mots est en lecture seule.

Le tableau suivant décrit les mots système de %SW20 à %SW27 :

Mots système	Adresse du nœud (numéro de l'esclave)		Description/contenu des mots
	Bit [15-8]	Bit [7-0]	
%SW20	2	1	Lorsque %SW2x prend la valeur suivante : ● = 0 => Tous les modules étaient manquants sur le bus CANopen au démarrage du maître CANopen(1). ● = 1 => Un module inattendu a été détecté sur le réseau. Ce module a été signalé comme "erroné" avant d'être supprimé du réseau. ● = 2 => Etat du nœud Operational (le module est opérationnel) : - "sans erreur". ● = 3 => Etat du nœud Operational (le module est opérationnel) : - "erroné". ● = 4 => Etat du nœud Pre-operational (le module est pré-opérationnel) : - modules attendus uniquement (tous les modules déclarés attendus dans le tableau de configuration) ; - le module peut être défini en mode Operational ; - "sans erreur". ● = 5 => Etat du nœud Pre-operational (le module est pré-opérationnel) : - modules attendus uniquement (tous les modules déclarés attendus dans le tableau de configuration) ; - le module peut être défini en mode Operational ; - "erroné".
%SW21	4	3	
%SW22	6	5	
%SW23	8	7	
%SW24	10	9	
%SW25	12	11	
%SW26	14	13	
%SW27	16	15	

Mots système	Adresse du nœud (numéro de l'esclave)		Description/contenu des mots
	Bit [15-8]	Bit [7-0]	
			• = 6 => Etat du nœud Pre-operational (le module est pré-opérationnel) : - modules attendus uniquement (tous les modules déclarés attendus dans le tableau de configuration) ; - le module a été détecté mais son état actuel ne permet pas de le définir en mode Operational ; - "sans erreur". • = 7 => Etat du nœud Pre-operational (le module est pré-opérationnel) : - modules attendus uniquement (tous les modules déclarés attendus dans le tableau de configuration) ; - le module a été détecté mais son état actuel ne permet pas de le définir en mode Operational ; - "erroné". • = 8 => Module incorrect (un module a été détecté avec des informations différentes concernant l'identité de l'équipement) : - "sans erreur". • = 9 => Module incorrect (un module a été détecté avec des informations différentes concernant l'identité du périphérique) : - "erroné". • = 10 => Erreur de configuration d'esclave (le module a répondu à une requête en écriture SDO de la table de commande SDO avec une confirmation d'erreur ou n'a pas respecté les règles du protocole SDO) : - "sans erreur". • = 11 => Erreur de configuration d'esclave : - "erroné". • = 12 => Module manquant / Erreur de contrôle de timeout / Timeout SDO (un module configuré n'est pas disponible, a disparu en cours de fonctionnement ou ne répond pas à l'accès SDO) : - "sans erreur".

Mots système	Adresse du nœud (numéro de l'esclave)		Description/contenu des mots
	Bit [15-8]	Bit [7-0]	
			• = 13 => Module manquant / Erreur de contrôle de timeout / Timeout SDO (un module configuré n'est pas disponible, a disparu en cours de fonctionnement ou ne répond pas à l'accès SDO) : - "erroné".(1) • = 14 => Module inattendu (un module absent du tableau de configuration a été détecté) : - "sans erreur". • = 15 => Module inattendu (un module absent du tableau de configuration a été détecté) : - "erroné".

Note :

1. Lorsque %SW2x prend la valeur suivante :
- = 0, pour un esclave, signifie que tous les esclaves sont manquants ;
- = 13, pour un esclave, signifie que cet esclave est manquant et qu'au moins un esclave est présent sur le bus CANopen.

Présentation de l'instruction CAN_CMD

Pour chaque programme utilisateur, l'instruction CAN_CMD permet à l'utilisateur de programmer son réseau et d'obtenir le diagnostic des esclaves. Les paramètres de l'instruction sont transmis par mots internes (mémoires) **%MW**x.

La syntaxe de l'instruction est la suivante :

CAN_CMDn **%MW**x**:**l

Légende :

Symbole	Description
n	Adresse d'expansion du module maître CANopen installé sur le bus Twido (1 à 7).
x	Numéro du premier mot interne (mémoire) transmis en paramètre.
l	Longueur de l'instruction en nombre de mots (2 ou 6).

Utilisation de l'instruction CAN_CMD

L'instruction CAN_CMD permet de programmer et de gérer le réseau CANopen et d'effectuer des vérifications de diagnostic des services esclaves individuels. Les paramètres de commande sont transmis via les mots mémoire %MWx.

Le tableau suivant décrit l'action de l'instruction CAN_CMD en fonction de la valeur des paramètres %MW(x), et %MW(x+5) si nécessaire :

<table>
<tr><th rowspan="2">%MWx</th><th colspan="2">%MWx+1</th><th colspan="2">%MWx+2</th><th colspan="2">%MWx+3</th><th colspan="2">%MWx+4</th><th colspan="2">%MWx+5</th><th rowspan="2">Action</th></tr>
<tr><th>Bit [15-8]</th><th>Bit [7-0]</th><th>Bit [15-8]</th><th>Bit [7-0]</th><th>Bit [15-8]</th><th>Bit [7-0]</th><th>Bit [15-8]</th><th>Bit [7-0]</th><th>Bit [15-8]</th><th>Bit [7-0]</th></tr>
<tr><td>1</td><td colspan="2">0</td><td colspan="8" rowspan="4">–</td><td>Réinitialise la communication CANopen</td></tr>
<tr><td>1</td><td colspan="2">1</td><td>Réinitialise les nœuds CANopen</td></tr>
<tr><td>2</td><td colspan="2">0</td><td>Passe du mode Operational au mode Pre-Operational</td></tr>
<tr><td>2</td><td colspan="2">1</td><td>Passe au mode Operational</td></tr>
<tr><td rowspan="6">3 ou 4</td><td colspan="2" rowspan="2"></td><td colspan="8" rowspan="2"></td><td>3 => Démarre la lecture de la commande SDO
4 => Démarre l'écriture de la commande SDO</td></tr>
<tr><td>Nœud = 1-16 => adresse du nœud</td></tr>
<tr><td colspan="2">Index</td><td colspan="8"></td><td>Index des objets PDO</td></tr>
<tr><td colspan="4"></td><td>Sub</td><td>Len</td><td colspan="4"></td><td>Sub = 0-255 => sous-index des objets
Len = longueur des données en octets</td></tr>
<tr><td colspan="6"></td><td colspan="2">Data 1</td><td colspan="2"></td><td>Charge selon la valeur du champ relatif à la longueur (Len)</td></tr>
<tr><td colspan="8"></td><td colspan="2">Data 2</td><td>Charge selon la valeur du champ relatif à la longueur (Len)</td></tr>
</table>

%MWx	%MWx+1		%MWx+2		%MWx+3		%MWx+4		%MWx+5		Action
	Bit [15-8]	Bit [7-0]	Bit [15-8]	Bit [7-0]	Bit [15-8]	Bit [7-0]	Bit [15-8]	Bit [7-0]	Bit [15-8]	Bit [7-0]	
5 (1)	Nœud										Réinitialise la communication CANopen pour un nœud
6 (1)	Nœud										Réinitialise les nœuds CANopen pour un nœud
7 (1)	Nœud										Passe du mode Operational au mode Pre-Operational pour un nœud
8 (1)	Nœud										Passe au mode Operational pour un nœud
(1) CAN_CMD disponible uniquement pour :Twido Extreme avec la version 4.0 ou ultérieure du microprogramme et pour le module maître CANopen TWDNCO1M avec la version 2.0 ou ultérieure du microprogramme.											

Note : L'état du bus est mis à jour à chaque scrutation de l'automate. Toutefois, le résultat de l'instruction CAN_CMD de lecture du bus n'est disponible qu'à la fin de la scrutation suivante.

Exemples de programmation de l'instruction CAN_CMD

Exemple 1 :

Pour forcer le passage du maître CANopen TWDNCO1M (situé à l'adresse 1 sur le bus d'expansion Twido) ou du maître de bus CANopen intégré de Twido Extreme en mode Init :

```
LD 1
[%MW0 := 16#0001]
[%MW1 := 16#0001]
LD %SW81:X3       (* Si aucune instruction CAN_CMD n'est *)
(* en cours, poursuivre *)
[CAN_CMD1 %MW0:2] (* Pour forcer le passage du maître CANopen
*)
(* en mode Init *)
----
LDN %SW81:X4      (* (facultatif) Pour savoir si l'instruction
CAN_CMD *)
(* a été correctement exécutée avant d'en *)
(* envoyer une autre. *)
```

Exemple 2 :

Pour lire la variable suivante : SDO_Slave:1_index:24576_sub-index:1_length:4

```
LD 1
[%MW6 := %MW4]    (* Enregistrer le résultat de la dernière *)
(* commande SDO *)
[%MW7 := %MW5]    (* Enregistrer le résultat de la dernière *)
(* commande SDO *)
----
LD %SW81:X3       (* Si aucune instruction CAN_CMD n'est *)
(* en cours, poursuivre *)
[%MW0 := 16#0003]
[%MW1 := 16#0001] (* Lecture SDO à l'adresse de nœud 1*)
[%MW2 := 16#6000] (* Accès à l'index numéro 24 576 *)
[%MW3 := 16#0104] (* Accès au sous-index numéro 1 *)
(* et à la valeur de longueur 4 *)
[CAN_CMD1 %MW0:6] (* Démarrer la commande SDO *)
```

Exemple 3 :

Pour écrire la variable suivante : SDO_Slave:1_index:24576_sub-index:1_length:4

```
LD 1
[%MW0 := 16#0004]
[%MW1 := 16#0001] (* Ecriture SDO à l'adresse de nœud 1*)
[%MW2 := 16#6000] (*Accès à l'index numéro 24 576 *)
[%MW3 := 16#0104] (* Accès au sous-index numéro 1 *)
(* et à la valeur de longueur 4 *)
[%MW4 := 16#1234] (* Valeur de Data 1 *)
[%MW5 := 16#1234] (* Valeur de Data 2 *)
----
LD %SW81:X3       (* Si aucune instruction CAN_CMD n'est *)
(* en cours, poursuivre *)
[CAN_CMD1 %MW0:6] (* Démarrer la commande SDO *)
```

Exemple 4 :

Pour forcer le passage du maître de bus CANopen intégré de Twido Extreme en mode Init :

```
LD 1
[%MW0 := 16#0005]   (*Type de commande 5 à 8*)
[%MW1 := 16#0001] (* Adresse du nœud *)
----
LD %SW81:X3        (* Si aucune instruction CAN_CMD n'est *)
(* en cours, poursuivre *)
[CAN_CMD1 %MW0:2]   (* Pour forcer le passage du maître de bus
CANopen *)
(* de Twido Extreme en mode Init *)
```

Echange sous tension CANopen pour les automates Twido

Echange sous tension CANopen

Le réseau CANopen équipé d'un automate Twido et d'un TWDNCO1M maître CANopen (version du microprogramme 2.0 ou ultérieure) prend en charge l'échange sous tension. Twido Extreme prend également en charge l'échange sous tension. Un échange sous tension consiste à déconnecter et à reconnecter un esclave CANopen sans mettre le maître hors/sous tension (ou sans le déconnecter de l'automate dans le cas d'un automate Twido Extreme).

Compatibilité de la fonctionnalité d'échange sous tension

Cette fonctionnalité est uniquement disponible si la supervision du nœud (Guard Time ou HeartBeat) a été configurée. Pour plus d'informations sur la configuration de la supervision du nœud, reportez-vous à la section *p. 278*.

Installation et configuration du bus de terrain CANJ1939

11

Présentation

Objet de ce chapitre

Ce chapitre décrit comment installer et configurer un bus de terrain CANJ1939 avec une base automate TWDLEDCK1 Twido Extreme.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
11.1	Vue d'ensemble du bus de terrain CANJ1939	298
11.2	Mise en œuvre du bus CANJ1939	305

11.1 Vue d'ensemble du bus de terrain CANJ1939

Présentation

Objet de cette section

Cette section propose des informations générales sur le bus de terrain CANJ1939 et présente la terminologie propre au CANJ1939 qui sera utilisée dans le reste du chapitre.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Base de connaissances CANJ1939	299
Numéro de groupe de paramètres et numéro de paramètre suspect CANJ19139	301
Identifiant CANJ1939	302
Communication sur un réseau CANJ1939	304

Base de connaissances CANJ1939

Introduction

La section suivante décrit les informations de base concernant les communications réseau CANJ1939.

Applications

CANJ1939 est un protocole de haut niveau pour les bus de terrain CAN développés par SAE (Society of Automotive Engineers). Il est destiné aux équipements utilisés dans de multiples domaines : agriculture, bâtiment, incendie/secours, foresterie, manutention de matériaux et équipements routiers ou non routiers.

Caractéristiques

CANJ1939 est un réseau de communication à haut débit conçu pour prendre en charge les fonctions de contrôle de boucle fermée en temps réel opérant entre les équipements électroniques distribués physiquement dans le véhicule.

CANJ1939 est capable d'exécuter toutes les fonctions de J1708/J1587, de prendre en charge le système de contrôle ainsi que toute application utilisant l'un ou l'autre des réseaux, ou les deux.

Note : Jl708/Jl587 est un réseau plus ancien largement utilisé qui garantit la simplicité de l'échange des informations, telles que les données de diagnostic, entre les équipements électroniques.

Bloc de commande électronique (ECU)

Un bloc de commande électronique (ECU) est un assemblage électronique informatisé à partir duquel les messages CANJ1939 peuvent être envoyés ou reçus.

Par la suite, dans ce document, les ECU sont appelés éléments CANJ1939.

Numéro de groupe de paramètres (PGN)

Dans les spécifications du CANJ1939, les paramètres liés ou similaires (signaux) sont regroupés dans un *groupe de paramètres (PG)*. Chaque groupe de paramètres est identifié par un numéro unique : son *Numéro de groupe de paramètres (PGN)* (reportez-vous à la *p. 301*).

Numéro de paramètre suspect (SPN)

Un *numéro de paramètre suspect (SPN)* est utilisé pour identifier un paramètre particulier (signal) associé à un élément CANJ1939 (reportez-vous à la *p. 301*).

Un SPN est un numéro 19 bits unique faisant partie d'un PGN.

Note : Une liste détaillée de tous les SPN et PGN est fournie par le comité SAE dans une feuille de calcul MS Excel (voir http://www.sae.org/).

Méthodes de communication

Il existe trois méthodes de communication principales au sein du CANJ1939 :

- Les communications poste à poste (voir *p. 304*) : le message est acheminé vers un équipement particulier et une adresse cible spécifique est incluse dans l'identifiant du message.
- Les communications de diffusion (voir *p. 304*) : le message est transmis sur le réseau sans être acheminé vers une destination spécifique. Cela permet à tous les équipements d'utiliser les données sans demander de messages de requête supplémentaires.
- Les communications propriétaires : non utilisées dans Twido Extreme

Sources supplémentaires

Pour plus d'informations sur CANJ1939, reportez-vous aux documents SAE suivants :

Numéro de référence du document	Nom du document
SAE J1939	Recommended Practice for a Serial Control and Communications Vehicle Network
SAE J1939-11	Physical Layer—250K Bits/s, Shielded Twisted Pair
SAE J1939-13	Off-Board Diagnostic Connector
SAE J1939-15	Reduced Physical Layer, 250K Bits/s, Un-Shielded Twisted Pair (UTP)
SAE J1939-21	Data Link Layer
SAE J1939-31	Network Layer
SAE J1939-71	Vehicle Application Layer
SAE J1939-73	Application Layer - Diagnostics
SAE J1939-81	Network Management Protocol

Vous pouvez également consulter les sites Internet suivants :

- Le site Internet de CAN In Automation (CIA) : http://www.can-cia.org/
- Le site Internet de Society of Automotive Engineers (SAE) : http://www.sae.org/

Numéro de groupe de paramètres et numéro de paramètre suspect CANJ19139

Introduction

Cette section présente les informations relatives à l'utilisation du numéro de groupe de paramètres et du numéro de paramètre suspect sur un réseau CANJ1939.

Groupe de paramètres (PG)

Un *Groupe de paramètres (PG)* est un ensemble de paramètres transmis au sein d'un message CANJ1939.

Les groupes de paramètres comprennent :

- des commandes ;
- des données ;
- des requêtes ;
- des acquittements ;
- des acquittements négatifs.

Le PG ne dépend pas du champ d'adresse source : toutes les sources sont autorisées à envoyer n'importe quel PG.

Numéro de groupe de paramètres (PGN)

Le numéro de groupe de paramètres (PGN) est une représentation 24 bits identifiant un PG particulier.

La structure PGN permet un total de 8 672 groupes de paramètres différents. Lors de la transmission d'un groupe de paramètres, le PGN est codé au niveau de l'identifiant CAN.

Numéro de paramètre suspect (SPN)

Un numéro de paramètre suspect (SPN) est un numéro 19 bits utilisé pour identifier un élément, un composant ou un paramètre particulier associé à un ECU (élément CANJ1939). Cette fonctionnalité est particulièrement utile pour le diagnostic. En effet, un ECU ayant détecté un défaut associé à un composant particulier, un capteur par exemple, peut transmettre un message d'avertissement identifiant le composant défectueux. Les SPN sont affectés par le comité de la SAE. Les 511 premiers SPN sont réservés et seront affectés plus tard afin de correspondre exactement au même numéro que celui de l'identifiant du paramètre (PID) de J1587.

Exemples de SPN

- Température du liquide de refroidissement du moteur
- Température du carburant
- Température de l'huile du moteur
- Température de l'huile du turbo
- Température du refroidisseur intermédiaire du moteur

Identifiant CANJ1939

Introduction

CANJ1939 propose une définition réseau complète en utilisant l'identifiant 29 bits (Trame étendue CAN de l'anglais "CAN Extended Frame") défini au sein du protocole CAN.

Note : CANJ1939 permet également aux équipements à identifiants 11 bits (Trame standard CAN de l'anglais "CAN Standard Frame") d'être utilisés au sein d'un même réseau, de définir tous les messages comme propriétaires et de permettre aux deux types d'équipements de coexister sans interférence.

Description de l'identifiant CANJ1939

La structure de la trame 29 bits de l'identifiant CANJ1939 est décrite de la manière suivante :

SOF	PRIORITE			R	DP	FORMAT PDU (PF)						SRR	IDE	PF (suite)		SPECIFIQUE PDU (PS)								ADRESSE SOURCE								RTR
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33

Les sections suivantes décrivent les différentes parties composant une trame CANJ1939.

Priorité

Le champ Priorité (3 bits) est utilisé pour déterminer la priorité d'un message lors d'un processus d'arbitrage :

- Une valeur égale à 000 représente la priorité la plus élevée. Les messages de priorité élevée sont utilisés pour les messages de contrôle haut débit.
- Une valeur égale à 111 représente la priorité la plus basse. Les messages de priorité basse sont utilisés pour les données non prioritaires.

Réservé (R)

Le bit R est réservé. Ce paramètre par défaut permet une utilisation ultérieure de ce bit pour d'autres finalités, tel que l'a défini le comité SAE (Society of Automotive Engineers).

Page de données (DP)

Le bit DP est utilisé comme sélecteur de page :

- La page 0 contient tous les messages actuellement définis.
- La page 1 est réservée à une utilisation ultérieure. Elle est conçue pour proposer une capacité d'expansion supplémentaire une fois que l'espace mémoire de la page 0 est saturé.

Format PDU (PF)

Le champ PF (8 bits) identifie un des deux formats de message qui peuvent être transmis :

- Si 0 ≤ PF ≤ 239, alors le format PDU1 est utilisé pour les méthodes de communication poste à poste (voir *p. 304*).
- Si 240 ≤ PF ≤ 255, alors le format PDU2 est utilisé pour les méthodes de communication de diffusion (voir *p. 304*).

Note : PDU signifie Unité de données du protocole de l'anglais "Protocol Data Unit".

Spécifique PDU (PS)

Le champ PS (8 bits) est dépendant de la valeur du PDU :

- Si le format PDU1 est utilisé, alors le champ PS contient une adresse cible (DA). 240 groupes de paramètres sont fournis dans chaque page de données du format PDU1.
- Si le format PDU2 est utilisé alors le champ PS contient une extension de groupe (GE). Le champ Extension de groupe, associé aux quatre bits les moins significatifs du champ Format PDU, fournit 4096 groupes de paramètres pour chaque page de données (DP).

Adresse source

Le champ Adresse source (8 bits) contient l'adresse unique de l'ECU (élément CANJ1939) transmettant le message.

Autres champs

- SOF (début de trame) : bit initial d'une trame CAN utilisé uniquement pour indiquer le début de la trame.
- SRR (Substituer requête distante) : ce bit est entièrement défini et contrôlé par CAN et par conséquent, n'est pas décrit ou modifié par CANJ1939.
- IDE (Bit d'extension d'identifiant) : ce bit est entièrement défini et contrôlé par CAN et par conséquent, n'est pas décrit ou modifié par CANJ1939.
- RTR (Requête de transmission distante) : cette fonction de CAN n'est pas utilisée dans CANJ1939.

Communication sur un réseau CANJ1939

Vue d'ensemble

Cette section explique les méthodes de détection de collision, d'affectation d'adresse et de communication sur un réseau CANJ1939.

Détection de collision

CANJ1939 utilise le protocole CAN qui permet aux équipements de transmettre un message sur le réseau lorsque le bus est inactif.

Les collisions sont évitées grâce au processus d'arbitrage qui a lieu lors de la transmission de l'identifiant (via un schéma d'arbitrage non destructeur). L'arbitrage permet aux messages hautement prioritaires de bénéficier de faibles délais d'attente car chaque équipement est sur un pied d'égalité en matière d'accès au réseau.

Affectation d'adresse

Dans un réseau donné, chaque adresse d'élément de réseau doit être unique (0 à 254, 255 étant disponible pour la diffusion). Les PGN ne sont pas dépendants de l'adresse source. Ainsi, un élément CANJ1939 peut transmettre n'importe quel message.

Diffusion

La plupart des messages utilisés sur le réseau CANJ1939 sont de type Diffusion. Cela signifie que les données sont transmises sur le réseau sans être acheminées vers une destination spécifique. Cela permet à tous les équipements d'utiliser ces données sans demander de messages de requête supplémentaires. Cela permet également des mises à jour logicielles ultérieures afin d'adapter facilement les nouveaux équipements (affectation d'adresses).

Poste à poste

Lorsqu'un message doit être acheminé vers un équipement particulier, une adresse cible spécifique peut être incluse dans l'identifiant du message.

11.2 Mise en œuvre du bus CANJ1939

Présentation

Introduction

Cette section décrit comment mettre en œuvre un bus de terrain CANJ1939 à l'aide d'une base automate Twido Extreme TWDLEDCK1.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble de la mise en œuvre du CANJ1939	306
Configuration matérielle	307
Méthode de configuration du CANJ1939	308
Boîtes de dialogue de configuration CANJ1939 (Elément, Réseau, Port)	310
Création ou suppression d'objets Transmettre/Recevoir CANJ1939	314
Affichage des objets Transmettre/Recevoir CANJ1939	322
Configuration de la diffusion pour le CANJ1939	325
Configuration poste à poste du CANJ1939	327
Configuration du CANJ1939 en mode Expert	329
Objets d'entrée/sortie CANJ1939	330
Requête d'une sortie PGN	333

Vue d'ensemble de la mise en œuvre du CANJ1939

Configuration matérielle et logicielle requise

Les solutions matérielles et logicielles ci-dessous sont requises pour la mise en œuvre d'un bus CANJ1939 sur le système de votre automate Twido :

Matériel	Configuration requise
Base automate Twido Extreme	Base Extreme • TWDLEDCK1
Connecteurs et câbles CANJ1939	Reportez-vous au Guide de référence du matériel de Twido Extreme.
Câbles de programmation pour l'automate Twido Extreme	**Câbles de programmation de PC vers automate : USB** • TSX CUSB485 et TWD XCAFJ010
Câble de programmation pour l'automate Twido Extreme	**Câble de programmation PC vers automate : Série** • VW3 A8106
Verrou électronique BlueTooth – pour permettre le transfert sans fil du PC à l'automate	**Adaptateur Bluetooth pour l'automate :** • VW3 A8114

Logiciel	Configuration requise
Logiciel de configuration de l'automate Twido	TwidoSuite version 2.0 ou supérieure

Procédure de mise en œuvre du réseau CANJ1939

La procédure ci-dessous est destinée à vous guider lors de l'installation, de la configuration et de l'utilisation du réseau CANJ1939 :

Etape	Description
1	Configuration matérielle
2	Méthode de configuration de CANJ1939
3	Boîtes de dialogue de configuration
4	Création d'objets Transmettre/Recevoir
5	Configuration de la diffusion
6	Configuration de poste à poste
7	Configuration du mode Expert
8	Objets E/S CANJ1939
9	Utilisation des requêtes SPN
Les sections suivantes résument toutes ces étapes.	

Configuration matérielle

Installation de l'automate Twido Extreme TWDLEDCK1

Pour installer l'automate Twido Extreme TWDLEDCK1, procédez comme suit :

Etape	Action	Description
1	Montage de l'automate Twido Extreme TWDLEDCK1	Installez l'automate Twido Extreme TWDLEDCK1 (horizontalement ou verticalement) à l'aide des trous de montage. Pour plus d'informations, reportez-vous à la rubrique Mounting Instructions, Twido Extreme Hardware Guide.
2	Câblage et raccordements du CANJ1939	Connectez l'unité d'alimentation et les équipements à l'aide du connecteur à 70 broches. Pour plus d'informations sur le câblage, reportez-vous à la section sur l'(Connectors Location, Contacts Location on the Connector, Twido Extreme Hardware Guide).

Méthode de configuration du CANJ1939

Vue d'ensemble

Cette section décrit la méthode de configuration générale du bus CANJ1939. Des détails supplémentaires sont proposés dans les sections suivantes.

La configuration du CANJ1939 est réalisée dans la fenêtre **Description** de TwidoSuite. Vous pouvez consulter les résultats de votre configuration à l'aide des commandes **Programme** → **Configurer le matériel, volet Configuration du module, onglet CANJ1939** (reportez-vous à la section *p. 322*).

Note :
1. La configuration du CANJ1939 peut être effectuée en mode local uniquement.
2. Aucune modification ne peut être apportée lorsque le Twido Extreme est connecté au PC. Seuls le nom du réseau, les noms d'élément et la case à cocher "Mode Expert" peuvent être modifiés en mode connecté.

Méthode de configuration du CANJ1939

Le tableau ci-dessous présente les différentes phases de mise en œuvre logicielle d'un réseau CANJ1939. Pour plus d'informations sur la procédure de création et de configuration d'un réseau CANJ1939 dans TwidoSuite, reportez-vous à la section (Création d'un réseau CANJ1939, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)).

Mode	Phase	Description
Local	Sélection du matériel	Dans l'espace de travail graphique de la fenêtre Description de TwidoSuite, créez un réseau CANJ1939 en sélectionnant une base automate TWDLEDCK1 Twido Extreme et le nombre souhaité d'équipements CANJ1939 dans le Catalogue de produits.
	Configuration du réseau CANJ1939	Configurez le réseau CANJ1939 dans la **fenêtre Description** en : ● allouant une adresse unique (0 à 254, 255 étant disponible pour la diffusion) (2) à chaque élément CANJ1939. Un même équipement CANJ1939 peut avoir plusieurs adresses. Dans ce cas, plusieurs éléments portant le même nom doivent être créés dans TwidoSuite, mais avec des adresses différentes. Pour plus d'informations sur ces boîtes de dialogue, reportez-vous à la section *p. 310*), ● connectant les éléments les uns aux autres dans l'espace de travail graphique. Reportez-vous à la section (Création d'un réseau CANJ1939, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). ● définissant des objets de message (recevoir ou transmettre) appropriés (8 octets de données maximum) pour chaque élément CANJ1939 correspondant. Pour plus d'informations, reportez-vous à la section *p. 314*.
	Affichage des objets de message CANJ1939	Il est possible d'afficher les objets de message auparavant définis dans la fenêtre **Programme → Configurer → Configurer le matériel (volet Configuration du module, onglet CANJ1939).** Pour plus d'informations sur ce volet de configuration, reportez-vous à la section *p. 322*.
	Attribution d'un nouveau nom aux objets de message (facultatif)	Les objets de message possèdent des noms descriptifs prédéfinis. Vous pouvez néanmoins modifier ces noms (32 caractères maximum) dans la fenêtre **Programme → Configurer → Configurer le matériel (volet Configuration du module, onglet CANJ1939).**
	Programmation	Programmation de la fonction CANJ1939
Mode connecté	Transfert	Transfert de l'application dans l'automate
	Mise au point	Mise au point de l'application à l'aide : ● de l'écran de mise au point permettant d'afficher les éléments (adresse, paramètres) et d'attribuer des adresses ; ● des écrans de diagnostic permettant d'identifier les défauts.

Note :

1. Un équipement CANJ1939 est représenté dans TwidoSuite par un ou plusieurs éléments CANJ1939. Un équipement (ou élément) peut également être désigné par le terme de bloc de commande électronique (ECU) (reportez-vous à la section *p. 299*).
2. Les adresses des éléments sont limitées aux valeurs 0 à 254. Reportez-vous à la section *p. 299*

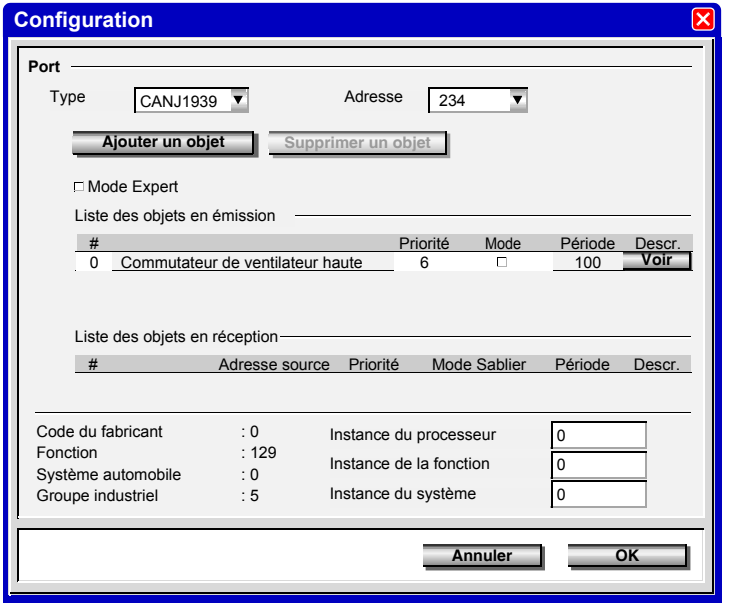
Boîtes de dialogue de configuration CANJ1939 (Elément, Réseau, Port)

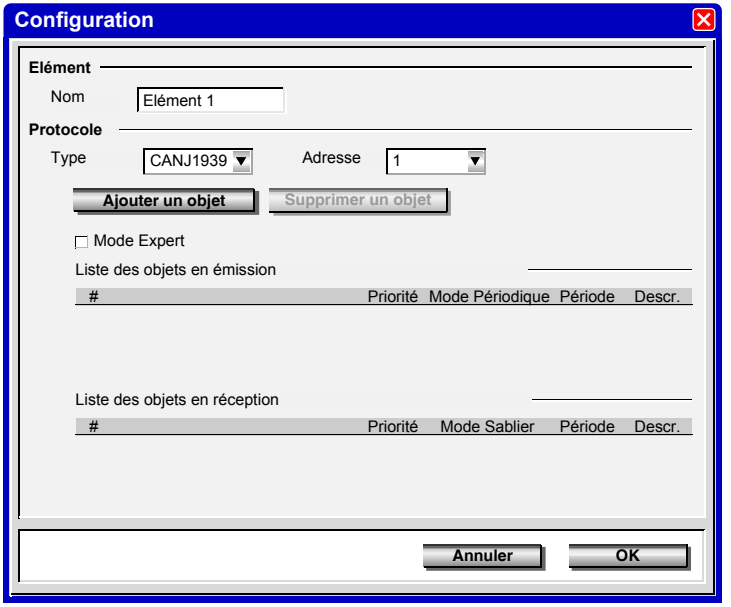
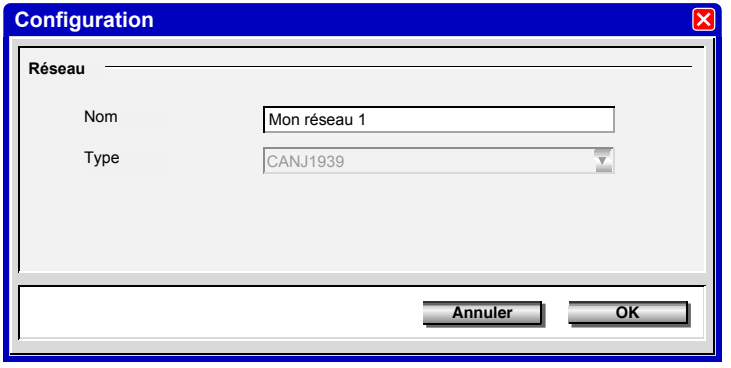
Vue d'ensemble

Cette section décrit les boîtes de dialogue Configuration CANJ1939 utilisées dans la fenêtre **Description** de TwidoSuite.

Configuration d'un élément/ réseau/port CANJ1939

Le tableau suivant indique comment configurer un élément, un réseau ou un port CANJ1939 que vous avez précédemment créé, et décrit les champs correspondants :

Etape	Action
1	Dans la fenêtre **Description**, passez la souris sur l'élément, le réseau ou le port jusqu'à ce que le curseur de configuration (tournevis) apparaisse, puis cliquez deux fois (ou cliquez avec le bouton droit de la souris et choisissez "Configurer"). **Résultat :** L'une des boîtes de dialogue de configuration suivantes s'ouvre (voir étapes2 3 et 4).
2	Boîte de dialogue de configuration de port : 

Etape	Action
3	Boîte de dialogue de configuration d'élément : 
4	Boîte de dialogue de configuration de réseau : 

Etape	Action
5	Les champs suivants peuvent être modifiés (facultatif) : • **Nom** : le nom par défaut est "Elément 1" (pour l'élément) et "Mon réseau 1" (pour le réseau). Ce champ est limité à 32 caractères. • **Adresse** : la valeur par défaut est 0. Les adresses des éléments sont limitées aux valeurs comprises entre 0 et 255. • Case à cocher **Mode Expert** : lorsqu'elle est cochée, les numéros PGN/SPN s'affichent. Par défaut, cette case n'est pas cochée. Si vous la cochez et que vous fermez la boîte de dialogue, elle sera décochée à la réouverture de la boîte de dialogue. • **#**: prend les valeurs de 0 à 31 et fournit un numéro d'identification simple pour les différents types de messages (reportez-vous à la section *p. 299*). Si deux objets de message ou plus proviennent du même groupe de paramètres, partageant ainsi le même #. Modifier la valeur # pour un objet applique automatiquement cette valeur aux autres. Le tableau est également réorganisé de manière à toujours afficher les objets par ordre de # croissant. • **Priorité** : prend les valeurs de 0 à 7 (0 étant la priorité la plus élevée). Ces valeurs sont préréglées en fonction de l'objet de message particulier (SPN) (reportez-vous à la section *p. 300*). Cependant, ces valeurs peuvent être modifiées. Modifier la priorité d'un objet de message applique automatiquement la même priorité à tous les objets de message partageant le même #. • Case à cocher **Mode périodique** (objets Transmettre) : lorsque cette case est cochée, le message est envoyé à intervalles réguliers selon la période (en ms) indiquée dans le champ Période. Si cette case n'est pas cochée, le champ Période ne peut pas être modifié. Modifier le mode périodique pour un objet de message applique automatiquement ce même mode à tous les objets de message partageant le même #. • Case à cocher **Mode sablier**(objets Recevoir) : lorsque cette case est cochée, la durée (en ms) indiquée dans le champ Période donne un temps d'inactivité. (Si Twido Extreme ne reçoit pas de message à l'issue de ce temps d'inactivité, une erreur de groupe de paramètres survient.) Les mots système %SW33 à %SW40 sont mis à jour. • **Période :** (en ms) prend les valeurs de 10 à 60 000 (la valeur par défaut est 100) et peut uniquement être modifiée si la case à cocher du mode périodique correspondant (objets Transmettre) ou Temps d'inactivité (objets Recevoir) est cochée. • **Descr :** cliquez sur ce bouton pour afficher une description de l'objet de message (SPN). • **Instance du processeur :**(1) indique à l'élément CANJ1939 quel automate va lui envoyer des messages. Les valeurs autorisées sont 0 et 1. • **Instance de la fonction :**(1) occurrence spécifique d'une fonction donnée dans un système automobile et un réseau donné. Si une seule instance d'une fonction spécifique existe dans un réseau, ce champ doit être réglé sur 0 pour définir cette instance comme la première. • **Instance du système automobile :**(1) occurrence spécifique d'un système de véhicule dans un réseau donné. Si une seule instance d'un système automobile spécifique existe dans un réseau, ce champ doit être réglé sur 0 pour définir cette instance comme la première. **Remarque :** (1)Toutes ces valeurs sont spécifiées dans SAE J1939.

Etape	Action
6	Les champs suivants sont en lecture seule : • **Code du fabricant :**(1) indique le fabricant chargé de la production du module de contrôle électronique. • **Fonction :**(1) indique la fonction principale de l'application automate. • **Système automobile :**(1) groupe de fonctions dans un réseau. • **Groupe industriel :**(1)indique un groupe d'industries. **Remarque :** (1)Toutes ces valeurs sont spécifiées dans SAE J1939.
7	Cliquez sur **Annuler** pour ignorer ou sur **OK** pour appliquer les modifications et fermer la boîte de dialogue.

Création ou suppression d'objets Transmettre/Recevoir CANJ1939

Vue d'ensemble

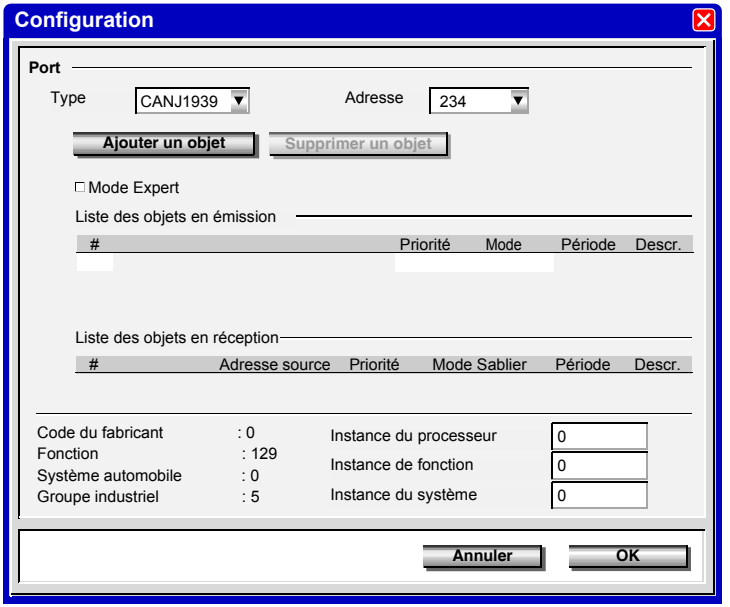
Les messages CANJ1939 peuvent être des objets **Transmettre** ou **Recevoir**. Ceux-ci sont créés à l'aide de la boîte de dialogue adéquate (élément ou port) accessible à partir de la fenêtre **Description**. Un objet Transmettre est un objet émis par Twido Extreme. Un objet Recevoir est un objet reçu par Twido Extreme.

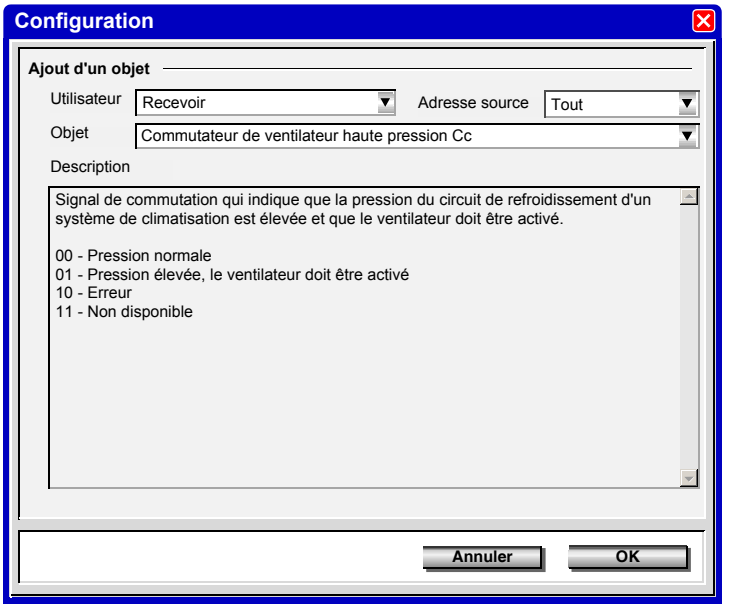
Cette section décrit la création et la suppression des objets Transmettre/Recevoir. Les messages de diffusion sont définis en ajoutant des objets à un port. Les messages poste à poste sont définis en ajoutant des objets à l'élément adéquat. Reportez-vous à la section *p. 304*.

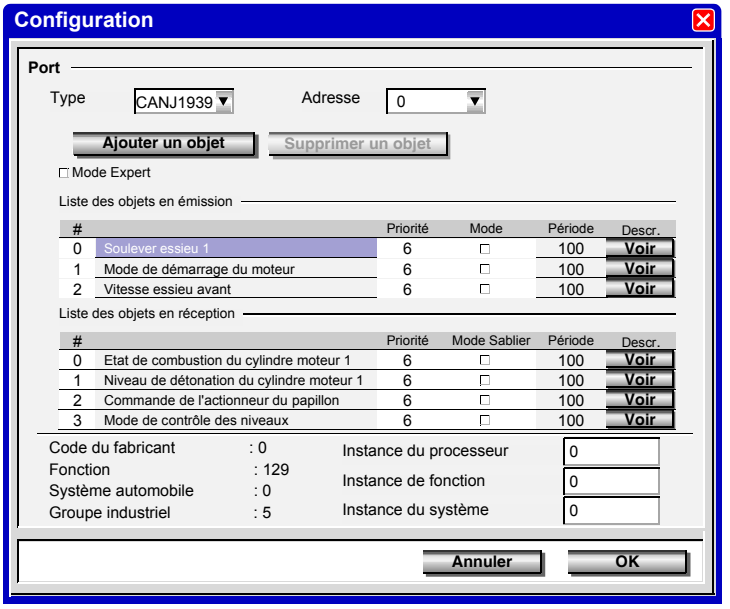
Note : Au niveau de TwidoSuite, vous pouvez configurer le réseau CANJ1939 pour les messages (de diffusion ou poste à poste) envoyés par l'automate Twido Extreme. TwidoSuite peut uniquement être utilisé pour configurer les automates Twido Extreme. Les produits autres que Twido raccordés au réseau CANJ1939 ne peuvent pas être configurés à l'aide de TwidoSuite.

Création d'objets Transmettre/ Recevoir

Les étapes suivantes décrivent la création des objets Transmettre/Recevoir.

Etape	Action
1	Dans la fenêtre **Description**, passez la souris sur l'élément adéquat (poste à poste) ou le port (diffusion) jusqu'à ce que le curseur de configuration (tournevis) apparaisse, puis cliquez deux fois (ou cliquez avec le bouton droit de la souris et choisissez sur "Configurer") pour ouvrir la boîte de dialogue de configuration. **Résultat :** La boîte de dialogue Configuration suivante s'ouvre (exemple avec la boîte de dialogue Port).  **Remarque :**Cet exemple concerne un message de diffusion. Les messages poste à poste sont créés de la même manière mais en configurant à la place un élément. Pour obtenir des exemples de boîte de dialogue de configuration d'un élément et davantage d'informations sur ces champs, reportez-vous à la section *p. 310*.

Etape	Action
2	Cliquez sur "**Ajouter un objet**". **Résultat :** La boîte de dialogue suivante s'affiche : 
3	Sélectionnez le type "**Recevoir**" ou "**Transmettre**", puis choisissez l'objet de message souhaité dans la liste **Type** (vous pouvez faire défiler cette liste **Type** à l'aide de la barre de défilement ou en tapant les trois premières lettres du nom de l'objet). Si l'objet choisi a déjà été créé sur un autre élément, port ou réseau CANJ1939 (avec le même type d'objet Recevoir/Transmettre), un message d'erreur s'affiche et vous ne pouvez pas ajouter cet objet. (Si vous cochez la case **Mode Expert**, les PGN/SPN correspondants s'affichent.(1)) Le **type** d'objet (Recevoir ou Transmettre) est fonction de ce que vous configurez (un élément ou un port) et du type de messagerie requise (diffusion ou poste à poste). Reportez-vous au Résumé des objets de message (voir *p. 318*) ci-dessous. Le champ "**Adresse source**" vous permet de décider si vous souhaitez gérer l'objet de message souhaité depuis tous les éléments CANJ1939 ou depuis un seul élément CANJ1939 en sélectionnant son adresse. Le champ "**Adresse source**" n'est pas disponible si vous avez sélectionné "**Transmettre**" dans la liste **Type**. (1) Les objets de message CANJ1939 sont regroupés en fonction de leur type (ou groupe de paramètres). Par conséquent, chaque objet de message est associé à un numéro de groupe de paramètres (PGN) hexadécimal et possède également sa propre identité hexadécimale unique désignée par le terme de numéro de paramètre suspect (SPN). Ainsi, un objet de message est souvent évoqué par son SPN. En outre, plusieurs SPN sont associés à un même PGN. Pour plus d'informations, reportez-vous à la section *p. 329*.

Etape	Action
4	Cliquez sur **Annuler** pour quitter ou sur **OK** pour ajouter l'objet sélectionné et fermer la boîte de dialogue "Ajouter un objet". **Résultat :** La boîte de dialogue Configuration (élément ou port) affiche les objets de message choisis. Exemple : 
5	Répétez les étapes 2 à 4 pour chaque objet de message que vous souhaitez créer pour cet élément ou port. Il n'est pas possible d'ajouter plus de 32 index de même type (Transmettre ou Recevoir) à un élément ou un port. Si vous dépassez cette limite, un message d'erreur s'affiche et l'ajout d'un nouvel objet est impossible.
6	Si besoin, résolvez les erreurs de conflit en supprimant des objets. Si la taille totale des objets de message choisis (SPN) dépasse les capacités de mémoire de TwidoSuite (cela dépend des SPN choisis, puisque leur taille varie), le message d'erreur suivant : "Trop de SPN définis sur le réseau" s'affiche et vous devez supprimer certains objets(2). (2) Si vous essayez d'ajouter un objet Transmettre (Recevoir) dépassant 8 octets et qu'un autre objet Transmettre (Recevoir) est déjà présent provenant du même groupe de paramètres (sur un élément ou un port), un message d'erreur de conflit s'affiche. Vous devez "Créer" (option par défaut) une nouvelle instance de ce groupe de paramètres comprenant votre objet de message. Vous avez également la possibilité de "Remplacer" (ce qui supprime l'objet de message en conflit du même groupe de paramètres) ou d'"Annuler" pour ignorer les modifications).

Etape	Action
7	(Facultatif) Vous pouvez manuellement modifier certains champs de la boîte de dialogue Configuration (élément ou port) tels que #, priorité, mode Périodique. Pour plus d'explications à propos de ces champs, reportez-vous à la section *p. 310*. Si besoin, modifiez manuellement les champs **#** pour affecter de nouveaux index aux groupes d'objets créés (PGN). Assurez-vous que les objets de message créés comportent des **index # consécutifs**. Pour plus d'informations sur le champ #, reportez-vous à la section Index d'objets de message ci-dessous.
8	Cliquez sur **Annuler** pour ignorer les modifications ou sur **OK** pour les appliquer et fermer la boîte de dialogue Configuration.

Résumé des objets de message

Ce tableau résume ce qu'implique l'ajout d'objets Transmettre/Recevoir sur un élément ou sur un port CANJ1939.

	Objet Transmettre ajouté	Objet Recevoir ajouté
Sur un Elément (poste à poste)	Twido Extreme envoie un message à un élément (ceci permet à l'élément de recevoir le message).	Twido Extreme prendra en charge ce message si cet élément l'envoie.
Sur le port CANJ1939 (diffusion)	Twido Extreme envoie un message à un élément configuré pour le recevoir (c'est-à-dire que l'élément a le même objet de message déclaré que l'objet Transmettre).	Twido Extreme prendra en charge ce message s'il apparaît sur le réseau. **Remarque** : Vous pouvez configurer Twido Extreme pour gérer ce message depuis tous les éléments CANJ1939 ou depuis un seul élément CANJ1939.

Les messages envoyés par d'autres équipements ne peuvent pas être configurés sur TwidoSuite.

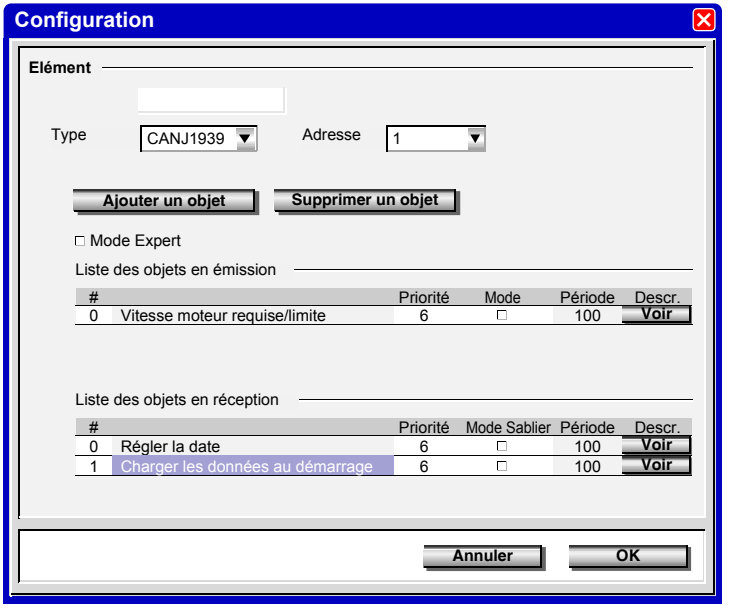
Index des objets de message

Le champ**#** est un index TwidoSuite pour les groupes d'objets créés (PGN) et prend les valeurs 0 à 31. Si deux ou plusieurs objets de message proviennent du même groupe de paramètres, ils partagent le même #. Ainsi, modifier la valeur # pour un objet applique automatiquement cette valeur aux autres. Le tableau est également réorganisé pour toujours afficher les objets par ordre de # croissant. Il est impossible de créer un objet de message avec un index # déjà utilisé par un autre élément ou port (TwidoSuite affiche un message d'erreur dans ce cas). Pour remédier à cela, modifiez manuellement le champ # et affectez un nouveau numéro d'index.

Remarque : Pour garantir que tous les SPN ajoutés seront ultérieurement accessibles dans la fenêtre **Programme** → **Configurer** → **Configure le matériel (volet Configuration du module, onglet CANJ1939)**, assurez-vous que les objets de message créés comportent des **index # consécutifs**. Par exemple, si vous ajoutez 6 SPN, ceux-ci sont automatiquement numérotés de #0 à #5. Si vous supprimez ensuite le SPN comportant l'index #3, les SPN #4 et #5 ne seront pas visibles dans l'écran de configuration. Pour résoudre ce problème, vous devez modifier manuellement les index #4 et #5 en les remplaçant par les index #3 et #4, afin d'obtenir une liste d'index consécutifs de #0 à #4.

Suppression d'objets Transmettre/ Recevoir

Les étapes suivantes décrivent la suppression des objets Transmettre/Recevoir.

Etape	Action
1	Dans la fenêtre **Description**, passez la souris sur l'élément adéquat ou le port jusqu'à ce que le curseur de configuration (tournevis) apparaisse, puis cliquez deux fois (ou cliquez avec le bouton droit de la souris et choisissez "Configurer") pour ouvrir la boîte de dialogue de configuration affichant les objets Transmettre/Recevoir précédemment créés. **Résultat :** La boîte de dialogue suivante s'affiche : 
2	Cliquez sur l'objet Recevoir/Transmettre à supprimer et cliquez sur "**Supprimer un objet**" ou appuyez sur la touche supprimer du clavier. (Si vous cochez la case Mode Expert, les PGN/SPN correspondants s'affichent.(1) (1) Les objets de message CANJ1939 sont regroupés en fonction de leur type (ou groupes de paramètres). Par conséquent, chaque objet de message est associé à un numéro de groupe de paramètres (PGN) hexadécimal et possède également sa propre identité hexadécimale unique désignée par le terme de numéro de paramètre suspect (SPN). Ainsi, un objet de message est souvent évoqué par son SPN. En outre, plusieurs SPN sont associés à un même PGN. Pour plus d'informations, reportez-vous à la section *p. 329*.
3	Répétez les étapes 1 à 2 pour chaque objet de message que vous souhaitez supprimer.

Etape	Action
4	Après avoir supprimé des objets de message, assurez-vous toujours que les **index # résultants sont consécutifs**. Ainsi, tous les SPN seront ultérieurement accessibles dans la fenêtre **Programme** → **Configurer** → **Configurer le logiciel (volet Configuration du module, onglet CANJ1939)**. Par exemple, si vous ajoutez 6 SPN, ceux-ci sont automatiquement numérotés de #0 à #5. Si vous supprimez ensuite le SPN comportant l'index #3, les SPN #4 et #5 ne seront pas visibles dans l'écran de configuration. Pour résoudre ce problème, vous devez modifier manuellement les index #4 et #5 en les remplaçant par les index #3 et #4, afin d'obtenir une liste d'index consécutifs de #0 à #4.
5	Cliquez sur **Annuler** pour ignorer ou sur **OK** pour supprimer l'objet sélectionné et fermer la boîte de dialogue Configuration.

Affichage des objets Transmettre/Recevoir CANJ1939

Vue d'ensemble

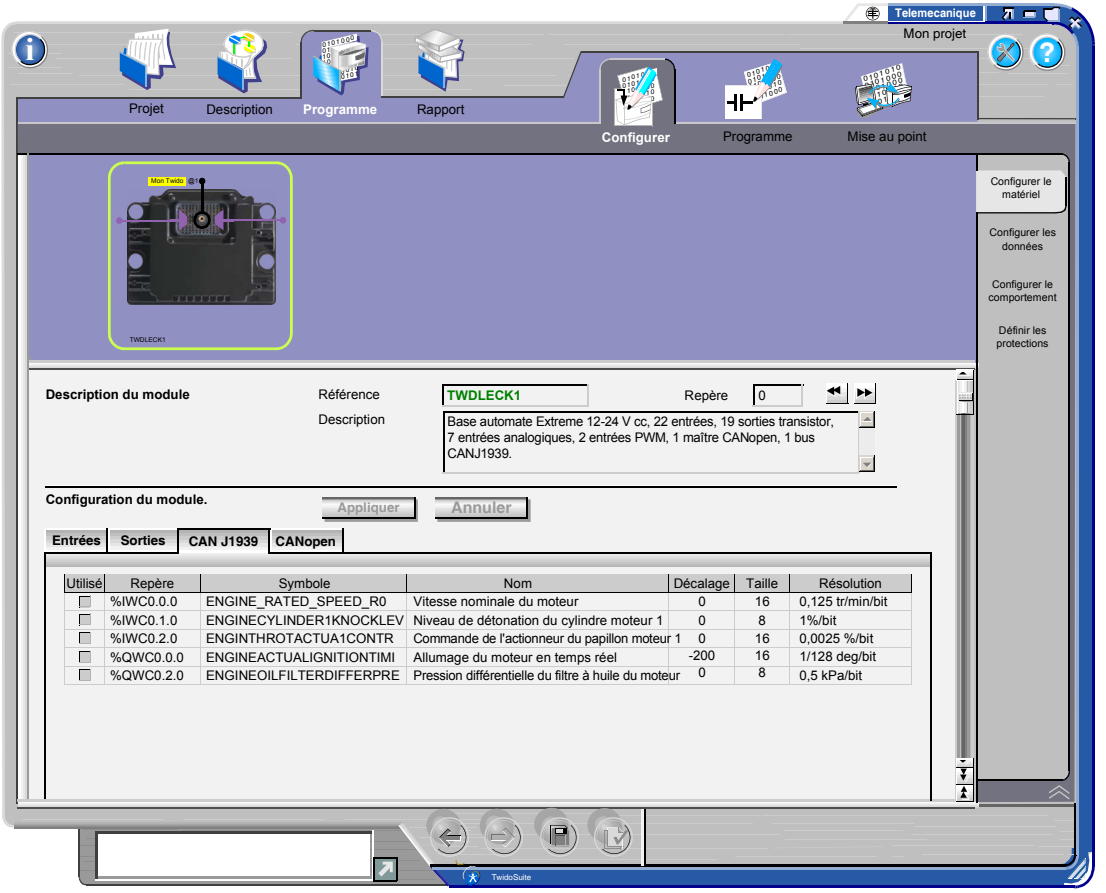
Cette section explique comment afficher les objets de message CANJ1939 précédemment définis dans la fenêtre **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** de TwidoSuite. Elle décrit également les champs se trouvant dans ce volet.

Note : Aucune configuration réelle ne peut être effectuée dans cette fenêtre.

Reportez-vous à la section *p. 308* pour obtenir des informations concernant la configuration générale et reportez-vous à la section *p. 314* pour ajouter et supprimer des objets de message.

Affichage des objets Transmettre/ Recevoir

Sélectionner les commandes **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** permet d'afficher les objets Transmettre/Recevoir précédemment créés, comme montré dans l'illustration suivante.


Telemecanique
Mon projet
Projet
Description
Programme
Rapport
Configurer
Programme
Mise au point
Configurer le matériel
Configurer les données
Configurer le comportement
Définir les protections
Description du module
Référence
TWDLECK1
Repère
0
Description
Base automate Extreme 12-24 V cc, 22 entrées, 19 sorties transistor, 7 entrées analogiques, 2 entrées PWM, 1 maître CANopen, 1 bus CANJ1939.
Configuration du module.
Appliquer
Annuler
Entrées
Sorties
CAN J1939
CANopen
Utilisé | Repère | Symbole | Nom | Décalage | Taille | Résolution
%IWC0.0.0 | ENGINE_RATED_SPEED_R0 | Vitesse nominale du moteur | 0 | 16 | 0,125 tr/min/bit
%IWC0.1.0 | ENGINECYLINDER1KNOCKLEV | Niveau de détonation du cylindre moteur 1 | 0 | 8 | 1%/bit
%IWC0.2.0 | ENGINTHROTACTUA1CONTR | Commande de l'actionneur du papillon moteur 1 | 0 | 16 | 0,0025 %/bit
%QWC0.0.0 | ENGINEACTUALIGNITIONTIMI | Allumage du moteur en temps réel | -200 | 16 | 1/128 deg/bit
%QWC0.2.0 | ENGINEOILFILTERDIFFERPRE | Pression différentielle du filtre à huile du moteur | 0 | 8 | 0,5 kPa/bit
TwidoSuite

Note : Pour garantir que tous les objets de message précédemment créés apparaissent dans ce volet, les **objets de message doivent être indexés de manière consécutive** dans la boîte de dialogue de configuration utilisée pour la Création ou suppression des objets Transmettre/Recevoir CANJ1939 (voir *p. 319*) accessible depuis la fenêtre **Description**.

Seul le champ Symbole peut être modifié dans ce volet. Si vous modifiez ce champ, il vous sera demandé si vous souhaitez appliquer ces modifications au moment de fermer la fenêtre. Si un objet avec un symbole modifié est supprimé ultérieurement, la modification apportée au symbole n'est pas enregistrée.

Pour plus d'informations sur le champ d'adresse, reportez-vous à la section *p. 330*.

Si un SPN est inférieur à 8 bits, ":Xi" est ajouté à l'adresse afin de donner le point de départ du SPN dans le mot.

Configuration de la diffusion pour le CANJ1939

Vue d'ensemble

Cette section décrit la configuration de la diffusion dans un réseau CANJ1939.

Messages de diffusion

Dans un réseau CANJ1939, chaque équipement (élément) possède au moins une adresse. Cependant, la plupart des messages sont diffusés et ne sont par conséquent pas dirigés vers une adresse de destination spécifique.

Les messages CANJ1939 sont constitués d'un identifiant codé sur 29 bits (pour plus d'informations, reportez-vous à la section *p. 302*), qui définit la priorité et l'expéditeur des messages, ainsi que les données qu'ils contiennent. Dans cet identifiant, les messages de diffusion se distinguent par un champ contenant une extension de groupe qui indique que le message doit être diffusé à tous les éléments du réseau et non à une adresse en particulier. Les objets de message de diffusion sont également désignés par le terme d'objets de type PDU2 (reportez-vous à la section *p. 302*).

Les messages de diffusion sont créés dans TwidoSuite en ajoutant des objets de message sur le **port** dans la fenêtre **Description** de TwidoSuite.

Messages de diffusion transmis par Twido Extreme

Le tableau ci-dessous indique comment créer un message CANJ1939 qui sera diffusé par Twido Extreme à tous les équipements réseau.

Etape	Action
1	Créez et configurez votre réseau CANJ1939 dans la fenêtre **Description** de TwidoSuite. Reportez-vous aux sections (Description des applications, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et (Création d'un réseau CANJ1939, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
2	Ajoutez les objets Transmettre souhaités **au port**. Ceux-ci seront diffusés à tous les éléments du réseau (reportez-vous à la section *p. 314*). Cela permet à tous les équipements d'utiliser les données contenues dans le message.
3	Ouvrez la fenêtre **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** pour afficher les objets de message que vous avez configurés à l'étape précédente. Reportez-vous à la section *p. 322*.

Messages de diffusion reçus par Twido Extreme

Twido Extreme peut être configuré afin de recevoir les messages diffusés sur le réseau CANJ1939 TwidoSuite peut être utilisé uniquement pour configurer les automates Twido Extreme. Les produits autres que Twido raccordés au réseau CANJ1939 ne peuvent pas être configurés à l'aide de TwidoSuite. Le tableau ci-dessous indique comment configurer Twido Extreme afin de recevoir les messages qui seront diffusés par l'élément CANJ1939.

Etape	Action
1	Créez et configurez le réseau CANJ1939 dans la fenêtre **Description** de TwidoSuite. Reportez-vous aux sections (Description des applications, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et (Création d'un réseau CANJ1939, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
2	Ajoutez les objets **Recevoir souhaités au port** (reportez-vous à la section (Création d'un réseau CANJ1939, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)). Ceux-ci seront reçus par Twido Extreme s'ils sont diffusés sur le réseau. **Remarque** : Vous pouvez configurer Twido Extreme pour gérer des messages provenant de tous les éléments CANJ1939 ou d'un seul élément CANJ1939. Reportez-vous à la section *p. 314*.
3	Ouvrez la fenêtre **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** pour afficher les objets de message que vous avez configurés à l'étape précédente. Reportez-vous à la section *p. 322*.

Configuration poste à poste du CANJ1939

Vue d'ensemble

Cette section décrit la configuration poste à poste dans un réseau CANJ1939.

Messages poste à poste

Les messages poste à poste sont définis dans un réseau CANJ1939 par un champ d'adresse de destination dans l'identifiant à 29 bits CANJ1939. Les objets de message poste à poste sont également désignés par le terme d'objets de type PDU1 (pour plus d'informations, reportez-vous à la *p. 302*).

Les messages poste à poste sont créés dans TwidoSuite en ajoutant des objets de message sur l'**élément** dans la fenêtre **Description** de TwidoSuite.

Messages poste à poste transmis par Twido Extreme et définis sur un élément

Le tableau ci-dessous indique comment créer un message CANJ1939 qui sera envoyé par Twido Extreme à une adresse spécifique.

Etape	Action
1	Créez et configurez votre réseau CANJ1939 dans la fenêtre **Description** de TwidoSuite. Reportez-vous aux sections (Description des applications, Aide en ligne TwidoSuite) et (Création d'un réseau CANJ1939, Aide en ligne TwidoSuite).
2	Ajoutez les objets **Transmettre** souhaités **à l'élément** (reportez-vous à la *p. 314*). Ce message sera envoyé par le Twido Extreme. Cela permet à cet élément de recevoir ce message lorsqu'il est transmis par le Twido Extreme.
3	Ouvrez la fenêtre **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** pour afficher les objets de message que vous avez configurés à l'étape précédente. reportez-vous à la *p. 322*.

Messages poste à poste reçus par Twido Extreme et définis sur un élément

Le Twido Extreme peut être configuré afin de recevoir les messages envoyés par un autre équipement (poste à poste) du réseau CANJ1939. TwidoSuite peut être utilisé uniquement pour configurer les automates Twido Extreme. Les produits autres que Twido raccordés au réseau CANJ1939 ne peuvent pas être configurés à l'aide de TwidoSuite. Le tableau ci-dessous indique comment configurer le Twido Extreme afin de recevoir les messages envoyés par un autre équipement (poste à poste) du réseau CANJ1939.

Etape	Action
1	Créez et configurez votre réseau CANJ1939 dans la fenêtre **Description** de TwidoSuite. Reportez-vous aux sections (Description des applications, Aide en ligne TwidoSuite) et (Création d'un réseau CANJ1939, Aide en ligne TwidoSuite).
2	Ajoutez les objets **Recevoir** souhaités **à l'élément** (reportez-vous à la *p. 314*). Ceux-ci seront reçus par Twido Extreme s'ils sont envoyés par cet élément.
3	Ouvrez la fenêtre **Programme** → **Configurer** → **Configurer le matériel (volet Configuration du module, onglet CANJ1939)** pour afficher les objets de message que vous avez configurés à l'étape précédente. reportez-vous à la *p. 322*.

Messages envoyés par d'autres équipements

Seuls les messages envoyés par le Twido Extreme peuvent être créés dans TwidoSuite. Les messages envoyés par d'autres éléments CANJ1939 (équipements) ne peuvent pas être configurés dans TwidoSuite. Ils doivent être créés dans le microprogramme de l'équipement.

Configuration du CANJ1939 en mode Expert

Vue d'ensemble

La configuration du CANJ1939 implique l'ajout des objets Transmettre ou Recevoir appropriés aux éléments du réseau. Reportez-vous à la section *p. 314*. Cela peut également être réalisé en mode Expert (en activant la case à cocher Mode Expert dans la boîte de dialogue Configuration). Cette section décrit l'utilisation de ce mode.

Sélection du mode Expert

Dans la fenêtre Description de TwidoSuite, cliquez deux fois sur le réseau, l'élément ou le port que vous souhaitez configurer pour ouvrir la boîte de dialogue Configuration. Activez la case à cocher **Mode Expert**.

Résultat : les boutons Ajouter/Supprimer un objet sont désormais appelés Ajouter/Supprimer un SPN. Les PGN/SPN correspondants sont révélés. Tous les objets qui sont ajoutés s'affichent avec leurs PGN et SPN.

PGN et SPN

Les objets de message CANJ1939 sont regroupés en fonction de leur type (ou groupes de paramètres). Par conséquent, chaque objet de message est associé à un numéro de groupe de paramètres (PGN) hexadécimal et possède également sa propre identité hexadécimale unique désignée par le terme de numéro de paramètre suspect (SPN). Ainsi, un objet de message est souvent évoqué par son SPN. En outre, plusieurs SPN sont associés à un même PGN. Pour plus d'informations, reportez-vous à la section *p. 301*.

Requête SPN

L'une des utilisations particulières du mode Expert consiste à demander des informations spécifiques à un élément CANJ1939 du réseau. Pour cela, un SPN Transmettre donné est ajouté à l'élément CANJ1939 et la valeur de ce SPN est attribuée au paramètre %IWC dans le programme en langage schéma à contacts ou instruction de liste. Pour obtenir un exemple, reportez-vous à la section *p. 333*.

Objets utilisateur CANJ1939

Les objets d'E/S sont les suivants :

- %IWC0.i.j, %QWC0.i.j (format mot - **il n'y a pas de bit d'E/S**)
- %IWCD0.i.j, %QWCD0.i.j (format mot double)
- %IWCF0.i.j, %QWC0.i.j (format flottant)

Où

- i est l'index du SPN (n° dans la boîte de dialogue de configuration de la description)
  i = 0,..., 31
- j est le numéro de voie (la position (en octets) du SPN)
  j = 0,...7 (pour les mots) ; j = 0,...,6 (autres cas)

Les E/S aux formats mot double et flottant (%IWCD, %QWCD et %IWCF, %QWCF) possèdent la même structure que les objets de mémoire interne aux formats mot double et flottant (%MD et %MF). Pour plus d'informations, reportez-vous à la section *p. 330*.

Objets d'entrée/sortie CANJ1939

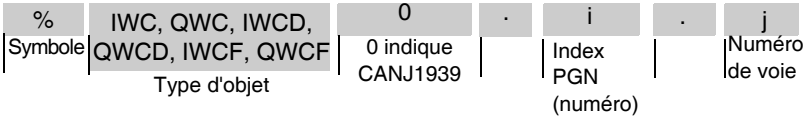
Introduction

Cette section décrit le repérage des entrées et sorties CANJ1939.

La syntaxe appliquée aux objets CANJ1939 est expliquée dans l'illustration suivante.

Illustration

Rappel des principes de repérage :



Valeurs spécifiques

Le tableau ci-dessous attribue des valeurs spécifiques aux objets CANJ1939 :

Elément	Valeurs	Commentaire
IWC	-	Image de l'entrée physique des PGN.
QWC	-	Image de la sortie physique des PGN.
IWCD	-	Même utilisation qu'IWC, mais au format double mot.
QWCD	-	Même utilisation que QWC, mais au format double mot.
IWCF	-	Même utilisation que IWC, mais au format flottant.
QWCF	-	Même utilisation que QWC, mais au format flottant.
0	0	Toujours O pour CANJ1939 (et toujours 1 pour CANopen).
i	0 à 31	Index du PGN (# dans la boîte de dialogue de configuration Description)
j	0 à 7 pour mot sinon 0 à 6	Numéro de voie (la position (en octets) du SPN)
Remarque : Si un SPN est inférieur à 8 bits, ":Xi" est ajouté au repère afin de donner le point de départ du SPN dans le mot. Ce repère ne peut pas être directement utilisé dans le programme Requête SPN. Cependant, la taille du SPN peut être utilisée dans un tel programme.		

Exemple

Le tableau suivant présente un exemple de repérage de CANJ1939 :

Objet d'E/S	Description
%IWC0.1.0	PGN 1, entrée de sous-index 0 du bus CANJ1939 Twido Extreme.

E/S mot double et flottant

Les mots longueur simple, double et flottants partagent la même zone mémoire. Ainsi, le mot flottant %IWCF0.i.1 et le mot double %IWCD0.i.1 correspondent aux mots longueur simple %IWC0.i.j et %IWC0.i.j+1 (le mot %IWC0.i.j contenant les bits de poids faible et le mot %IWC0.i.j+1 les bits de poids fort du mot %IWCF0.i.j).

Le tableau ci-dessous présente le recouvrement de l'allocation de mémoire pour les mots longueur simple et double flottants de type %IWC. La même chose s'applique à %QWC.

<table>
<tr><th colspan="2">Format double et flottant</th><th>Format mot simple</th></tr>
<tr><td>Repère pair</td><td>Repère impair</td><td></td></tr>
<tr><td rowspan="2">%IWCD0.i.0/
%IWCF0.i.0</td><td>...</td><td>%IWC0.i.0</td></tr>
<tr><td rowspan="3">%IWCD0.i.1/%IWCF0.i.1</td><td>%IWC0.i.1</td></tr>
<tr><td rowspan="2">%IWCD0.i.2/
%IWCF0.i.2</td><td>%IWC0.i.2</td></tr>
<tr><td rowspan="3">%IWCD0.i.3/%IWCF0.i.3</td><td>%IWC0.i.3</td></tr>
<tr><td rowspan="2">%IWCD0.i.4/
%IWCF0.i.4</td><td>%IWC0.i.4</td></tr>
<tr><td rowspan="3">%IWCD0.i.5/%IWCF0.i.5</td><td>%IWC0.i.5</td></tr>
<tr><td rowspan="2">%IWCD0.i.6/
%IWCF0.i.6</td><td>%IWC0.i.6</td></tr>
<tr><td>...</td><td>%IWC0.i.7</td></tr>
</table>

Note : Il n'y a aucune instruction spéciale (telle que CAN_CMD pour CANopen) dans CANJ1939.

Mots système réservés CANJ1939

Les mots système sont réservés aux informations relatives à l'état.

%SW80 contient 16 bits de mémoire (format du mot) pour donner les informations relatives à l'état du port CANJ1939.

%SW80 est décrit de la manière suivante :
- Bit [0] Erreur d'initialisation - Repère perdu à l'occasion d'une réclamation
- Bit [1] Erreur d'initialisation - Impossible de réclamer un repère
- Bit [2] Etat passif d'erreur sur le port
- Bit [3] Etat du bus désactivé sur le port

%SW33 à %SW40 fournissent des informations sur l'état de 32 objets entrée/sortie PGN.

Le tableau suivant présente ces mots système entrée/sortie

%SWx	Numéro de l'objet PGN		Contenu
%SW33	3 -2	1 - 0	4 bits par PGN : 0 = Etat normal 1 = PGN reçu sans erreur 2 = force l'écriture sur la sortie PGN 4 = erreur PGN (entrée ou sortie) Il existe trois types d'erreur PGN : ● erreur à la réception du PGN ; ● temps d'inactivité à la réception du PGN ; ● erreur à l'émission du PGN.
%SW34	7 - 6	5 - 4	
%SW35	11- 10	9 - 8	
%SW36	15 -14	13- 12	
%SW37	19 - 18	17 - 16	
%SW38	23 - 22	21 - 20	
%SW39	27 - 26	25 -24	
%SW40	31 -30	29 - 28	

Par exemple, pour forcer l'écriture sur une sortie PGN pour un PGN particulier, définissez bit 2 dans le %SW correspondant.

Programmation CANJ1939

Les programmes en langage schéma à contacts ou liste d'instructions sont créés de manière identique sur d'autres applications Twido. Reportez-vous aux sections *p. 457* et *p. 483*. Lors de l'utilisation du navigateur de données dans l'éditeur du programme en langage schéma à contacts/liste d'instructions, sélectionnez le type d'objet adéquat (%IWC0, %QWC0, %IWCD0, %QWCD0, %IWCF0 et %QWCF0) dans le menu Repère. Si Twido Extreme a été dernièrement remplacé par une base ne prenant pas charge le protocole CANJ1939, ces entrées doivent être supprimées du programme. Elles ne sont plus disponibles dans la liste Repère du navigateur de données.

Messages d'erreur CANJ19393

Les conditions suivantes engendrent des messages d'erreur dans le programme relatif aux objets d'E/S CANJ1939 :
- Lors de la saisie d'un objet (%IWC0, %QWC0, %IWCD0, %QWCD0, %IWCF0 et %QWCF0), le numéro SPN i (0 à 31) ou le numéro de voie j (0 à 7) ne se situe pas dans la plage.
- L'automate choisi dans la fenêtre Description ne prend pas en charge le protocole CANJ1939.
- Il n'y a pas de SPN défini au repère i et à la voie j.

Requête d'une sortie PGN

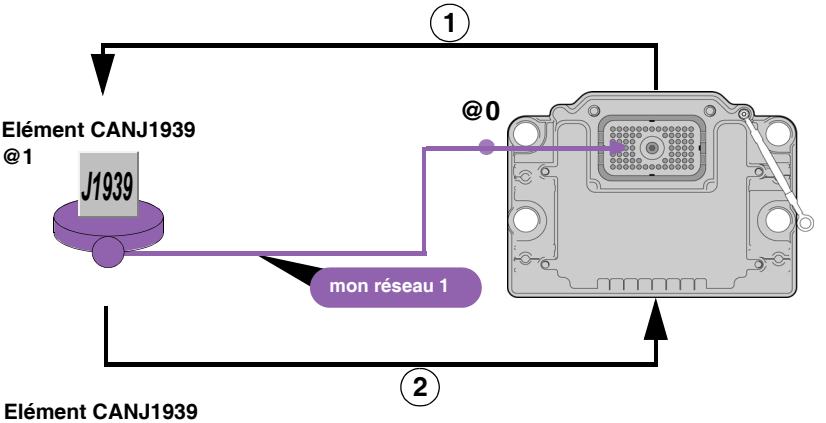
Introduction

Un PGN est envoyé par un élément CANJ1939 à chaque fois que sa valeur change. Pour forcer une sortie PGN, utilisez l'une des deux méthodes suivantes :

- Le SPN appelé RQST est utilisé par Twido Extreme pour demander un PGN à partir d'un élément CANJ1939.
- Le mot système %SW33 est utilisé pour forcer une sortie PGN.

Processus d'une requête PGN

Le schéma suivant illustre une requête PGN par le biais d'un exemple. Dans cet exemple, Twido Extreme envoie une requête à l'élément CANJ1939 pour le groupe de paramètres Exhaust Temperature. Ce processus est expliqué dans le tableau suivant :



Elément CANJ1939

Liste des objets en émission

Numéro de groupe de paramètre (RQST)

Liste des objets en réception

Température des gaz d'échappement du moteur - Collecteur droit

Remarque : Température des gaz d'échappement du moteur SPN - Collecteur droit fait partie du PGN Température d'échappement global.

Le tableau suivant présente le processus d'une requête PGN.

Etape	Description
1	Twido Extreme envoie le SPN RQST à un élément CANJ1939 pour demander la valeur de la température d'échappement PGN
2	L'élément CANJ1939 envoie la valeur du PGN demandé (température d'échappement)

Utilisation de SPN RQST

Pour utiliser la méthode SPN RQST, configurez le réseau CANJ1939 tel qu'indiqué ci-dessous dans l'exemple :

Etape	Action	Fenêtre dans TwidoSuite
1	Créez un réseau CANJ1939 avec un élément CANJ1939. Pour plus d'informations sur la création d'un réseau CANJ1939, reportez-vous à la rubrique Création d'un réseau CANJ1939, Aide en ligne TwidoSuite.	Description
2	Au niveau de l'élément CANJ1939 : ● Ajoutez le numéro de groupe de paramètres (RQST) émis par SPN. Pour plus d'informations sur la configuration adéquate d'un SPN (Recevoir ou Transmettre), reportez-vous à la *p. 314*. ● Ajoutez le SPN Recevoir que vous souhaitez demander (par exemple Température des gaz d'échappement du moteur SPN - Collecteur droit). Pour plus d'informations sur la configuration adéquate d'un SPN (Recevoir ou Transmettre), reportez-vous à la *p. 314*.	Description
3	Au niveau de l'élément CANJ1939 : **1.** Cliquez avec le bouton droit de la souris et sélectionnez Configurer. **2.** Activez la case à cocher **Mode Expert**. **3.** Enregistrez le PGN que vous souhaitez demander (par exemple, le numéro du groupe de paramètres Exhaust Temperature est 65031).	Description
4	Enregistrez le repère du mot d'échange en sortie (%QWCD0.y.z) affecté à SPN RQST que vous avez ajouté à l'étape précédente.	Programme → Configurer → Configurer le matériel → onglet CANJ1939 → champ Repère
5	Affectez le PGN à forcer au mot d'échange en sortie %QWCD0.y.z enregistré. **Exemple :** pour faire une requête sur PG Exhaust Temperature (PGN=65031(16#FE07)), utilisez la liste d'instructions : `%QWC0.0.0 := 16#FE07` **Remarque :** Vous pouvez également régler %QWCD0.y.z connecté à l'aide de la table d'animation.	Programme → Programme → Editer le programme
6	Consultez la valeur du paramètre forcé dans le mot d'échange en entrée %IWCD0.y.z. connecté en : ● utilisant la table d'animation (reportez-vous à la rubrique Déboguage en mode connecté via les tables d'animation de programme, Aide en ligne TwidoSuite) ; ● connectant %IWCD0.y.z sur une sortie.	
7	Calculez la valeur réelle du paramètre forcé : $\text{Actual value} = \text{Value of } \%\text{IWCD0.y.z} \times \text{resolution} + \text{offset}$	
Remarque :	**Si le paramètre demandé n'est pas disponible, l'élément CANJ1939 envoie un message de non prise en compte NACK (de l'anglais Non-ACKnowledgement) à Twido Extreme.**	

Utilisation du mot système %SW33

Pour utiliser la méthode du mot système %SW33, suivez la procédure illustrée dans l'exemple d'application ci-dessous :

Etape	Action	Fenêtre dans TwidoSuite
1	Au niveau de l'élément CANJ1939 sur lequel vous souhaitez forcer la sortie PGN : **1.** Cliquez avec le bouton droit de la souris et sélectionnez Configurer. **2.** Activez la case à cocher **Mode Expert**. **3.** Enregistrez le PGN que vous souhaitez forcer.	Description
2	Enregistrez le repère du mot d'échange en sortie (%QWCD0.y.z) affecté au SPN que vous souhaitez forcer.	Programme → Configurer → Configurer le matériel → onglet CANJ1939 → champ Repère
3	Réglez le mot système %SW33 sur 2 (voir exemple ci-dessous). **Remarque :** Vous pouvez également régler %QWCD0.y.z connecté à l'aide de la table d'animation.	Programme → Programme → Editer le programme
4	Affectez le PGN à forcer au mot d'échange en sortie %QWCD0.y.z enregistré (voir exemple ci-dessous). **Remarque :** Vous pouvez également régler %QWCD0.y.z connecté à l'aide de la table d'animation.	Programme → Programme → Editer le programme
5	Consultez la valeur du paramètre demandé dans le mot d'échange en entrée %IWCD0.y.z. en : ● utilisant la table d'animation (reportez-vous à la rubrique Déboguage en mode connecté via les tables d'animation de programme, Aide en ligne TwidoSuite) ; ● connectant %IWCD0.y.z sur une sortie.	
6	Calculez la valeur réelle du paramètre demandé : $\text{Actual value} = \text{Value of \%IWCD0.y.z} \times \text{resolution} + \text{offset}$	

Exemple du mot système %SW33

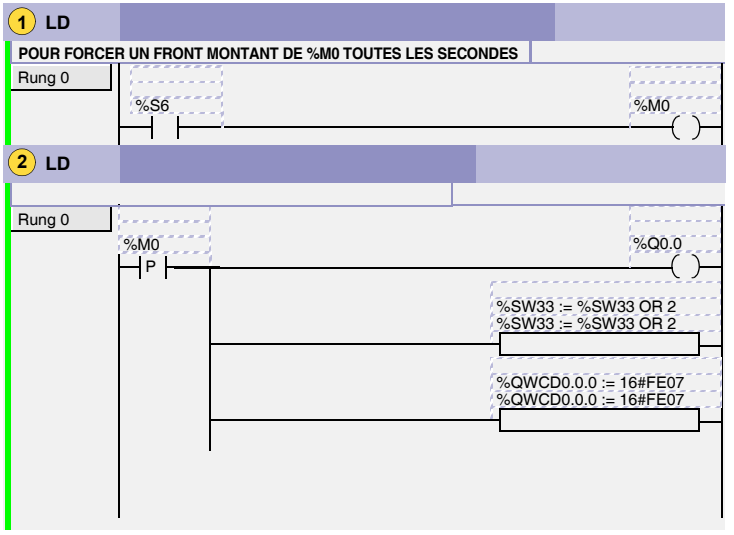
Cet exemple explique comment forcer une sortie PG Exhaust Temperature toutes les secondes :

Hypothèses :

- Vous souhaitez forcer la transmission du PG Exhaust Temperature (PGN=65031(16#FE07)).
- Le mot d'échange en sortie %QWCD0.0.0 est affecté au PGN 65031.

```
(*pour forcer un front montant de %M0 toutes les secondes*)
LD    %S6
ST    %M0
LDR   %M0
ST    %Q0.0.0
[ %SW33 := %SW33 OR 2 ](*Sur chaque front montant de %M0*)
(*mot système %SW33 réglé sur 2*)
[ %QWCD0.0.0 := 16#FE07 ](*PG Exhaust Temperature est
affecté*)
(*au mot d'échange en sortie %QWCD0.0.0*)
```

Même exemple ci-dessous avec un programme en langage schéma à contacts :



Configuration de la passerelle Ethernet TwidoPort

12

Aperçu

Objet de ce chapitre

Ce chapitre fournit des informations sur la configuration logicielle du module de passerelle Ethernet TwidoPort ConneXium.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
12.1	Configuration normale et connexion du module TwidoPort	338
12.2	Configuration Telnet de TwidoPort	348
12.3	Fonctions de communication	363

12.1 Configuration normale et connexion du module TwidoPort

Présentation

Objet de cette section

Ce sous-chapitre explique comment configurer normalement le module TwidoPort ConneXium avec l'application TwidoSuite et contient également des informations sur la configuration BootP et la connexion du module.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration normale avec TwidoSuite	339
Configuration BootP	347

Configuration normale avec TwidoSuite

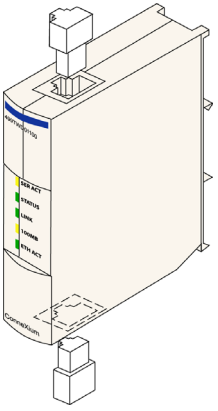
Avant-propos

Configurez TwidoPort à l'aide des instructions suivantes :

Note : Fonction Plug-and-Play
Lorsque vous configurez le module TwidoPort avec TwidoSuite, la configuration IP du module est stockée dans l'automate Twido. Dès lors, l'équipe de maintenance peut échanger les modules TwidoPort sans effectuer de configuration supplémentaire. Pour utiliser la fonctionnalité Plug-and-Play, utilisez TwidoSuite et mettez à niveau le microprogramme Twido avec la version 3.4 ou supérieure.

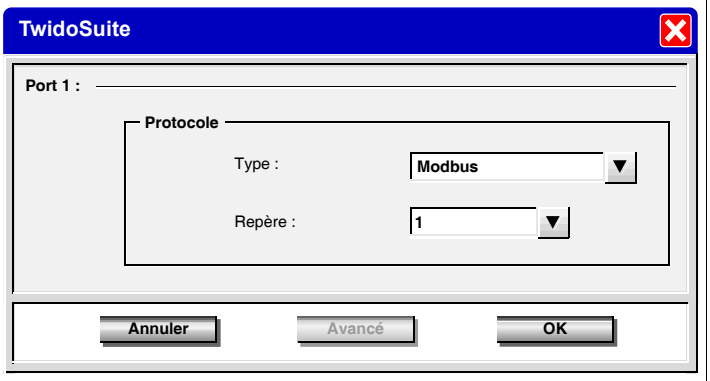
Installation du module TwidoPort 499T WD01100

Pour installer le module TwidoPort sur un automate Twido (montage sur panneau ou sur rail DIN) et le connecter au bus interne de l'automate, procédez comme suit :

Etape	Description	Action
1	Préparation de l'installation	Consultez le *guide de référence du matériel d'automates programmables Twido (TWD USE 10AE)*, pour obtenir des instructions sur : ● les positions de montage correctes des modules Twido ; ● l'ajout de composants Twido sur un rail DIN, ou leur suppression ; ● le montage direct sur un panneau ; ● les dégagements minimaux des modules dans un panneau de commande.
2	Montage du module TwidoPort 499TWD01100	Installez le module sur un panneau ou sur un rail DIN. Pour plus d'informations, reportez-vous à la section Installation du module d'interface Ethernet TwidoPort.
3	Prise de terre de protection (PE)	Connectez un fil de terre au bornier à vis M3 en bas du module TwidoPort.
4	Connexions Ethernet et série **Connecteur supérieur : Twido (série)**  **Connecteur inférieur : Etherne (câble croisé ou droit)**	Connectez l'extrémité avec prise modulaire du câble TwidoPort/Twido (fourni) au port série du module TwidoPort et l'autre extrémité au port série RS485 de l'automate Twido. Reliez le connecteur RJ45 d'un câble réseau Ethernet standard (non fourni) au port Ethernet du module TwidoPort.

Déclaration du module TwidoPort 499TWD01100

Le tableau ci-dessous présente les différentes étapes nécessaires à la déclaration d'un module TwidoPort 499TWD01100.

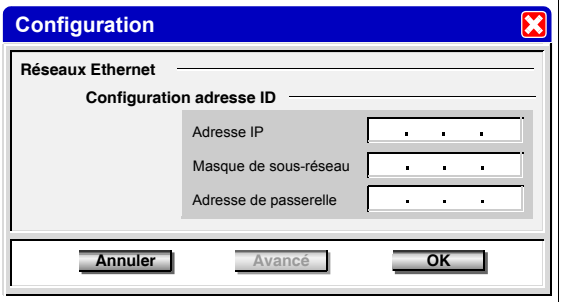
Etape	Action	Commentaire
1	Sélectionnez le **port 1** (ou le **port 2** s'il est installé) à configurer dans la fenêtre Description.	Reportez-vous à la section (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
2	Configurez la boîte de dialogue Caractéristiques (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) qui s'affiche, en procédant comme suit :	
3	Dans la zone Type de protocole, sélectionnez **Modbus**.	
4	Sélectionnez **Description** dans l'interface TwidoSuite.	Reportez-vous à la section (Description des applications, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
Remarque 1	**Tout port Modbus RS485 situé sur l'automate Twido peut être utilisé.**	
Remarque 2	**Pour obtenir la détection automatique de la vitesse de modulation initiale la plus rapide, choisissez 19200-8-N-1, en paramétrant l'adresse Modbus de Twido sur 1.**	
Remarque 3	**Le module TwidoPort 499TWD01100 doit également être configuré (reportez-vous à la section *p. 343* ci-dessous) et lors de la Configuration d'une connexion Ethernet dans TwidoSuite (voir *p. 345*), la valeur par défaut "Direct" de P-Unit/Repère doit être définie sur l'adresse du port Modbus (par exemple, sur 1 ou sur toute adresse préalablement utilisée pour l'automate).**	

Etape	Action	Commentaire
5	Affichez le catalogue des produits et sélectionnez un module **499TWD01100** à ajouter à la description du système.	Reportez-vous à la section (Positionnement d'un module, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). Vous pouvez continuer à ajouter autant de modules optionnels que vous le souhaitez, du moment qu'ils sont pris en charge par l'automate Twido. **Remarque :** Vous ne pouvez connecter qu'un seul module TwidoPort 499TWD01100.
Remarque 1	**Tout port Modbus RS485 situé sur l'automate Twido peut être utilisé.**	
Remarque 2	**Pour obtenir la détection automatique de la vitesse de modulation initiale la plus rapide, choisissez 19200-8-N-1, en paramétrant l'adresse Modbus de Twido sur 1.**	
Remarque 3	**Le module TwidoPort 499TWD01100 doit également être configuré (reportez-vous à la section *p. 343* ci-dessous) et lors de la Configuration d'une connexion Ethernet dans TwidoSuite (voir *p. 345*), la valeur par défaut "Direct" de P-Unit/Repère doit être définie sur l'adresse du port Modbus (par exemple, sur 1 ou sur toute adresse préalablement utilisée pour l'automate).**	

Configuration du module TwidoPort 499TWD01100

Note : Vous pouvez configurer les paramètres Ethernet du module TwidoPort uniquement lorsque le programme d'application TwidoSuite est en mode local.

Pour configurer les paramètres Ethernet du module TwidoPort, suivez les instructions de la procédure ci-dessous :

Etape	Action	Commentaire
Avant-propos	Pour obtenir plus d'informations sur les paramètres IP (adresse IP, masque de sous-réseau et adresse de passerelle), reportez-vous aux sections (Adressage IP, Twido, Guide de programmation) et (Affectation d'adresses IP, Twido, Guide de programmation).	
1	Sélectionnez le module TwidoPort 499TWD01100 pour configurer les paramètres IP de TwidoPort (reportez-vous à la section (Configuration d'un objet, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)). **Résultat :** La fenêtre **TwidoPort Configuration** apparaît à l'écran, comme illustré dans la sous-section suivante.	**Résultat :** La boîte de dialogue **Configuration Ethernet** apparaît, comme illustré ci-dessous : 
2	Entrez l'**adresse IP** statique de TwidoPort en notation décimale séparée par des points. (Voir les *remarques 1 et 2*.)	**Attention :** Pour une bonne communication entre les équipements, les adresses IP du PC exécutant l'application TwidoSuite et du module TwidoPort doivent partager le même ID réseau.
Remarque 1	**Consultez votre administrateur réseau ou système pour obtenir les paramètres IP valides du réseau.**	
Remarque 2	**Pour une bonne communication sur le réseau, les équipements connectés doivent avoir une adresse IP unique. Lorsque TwidoPort est connecté au réseau, il recherche la présence de doublons d'adresse IP. Si un doublon est détecté sur le réseau, le voyant d'état clignote 4 fois à intervalles réguliers. Vous devez alors renseigner ce champ avec une adresse IP unique.**	
Remarque 3	**Utilisez le masque de sous-réseau par défaut, à moins que le module TwidoPort ne comporte des exigences particulières en matière de sous-réseau.**	
Remarque 4	**S'il n'existe aucune passerelle sur le réseau, saisissez simplement l'adresse IP de TwidoPort dans le champ Adresse de la passerelle.**	

Etape	Action	Commentaire
3	Saisissez le **masque de sous-réseau** valide affecté au module TwidoPort par l'administrateur réseau. Notez que ce champ est obligatoire. (Voir les *remarques 1 et 3*.)	**Attention :** Pour une bonne communication entre les équipements, le masque de sous-réseau configuré sur le PC exécutant l'application TwidoSuite et celui du module TwidoPort doivent correspondre. Par défaut, l'application TwidoSuite calcule automatiquement et affiche le masque de sous-réseau par défaut en fonction de la classe de l'adresse IP définie dans le champ précédent. Selon la catégorie d'adresse IP réseau de TwidoPort, les valeurs du masque de sous-réseau par défaut respectent la règle suivante : Réseau de classe A -> Masque de sous-réseau par défaut : 255.0.0.0 Réseau de classe B -> Masque de sous-réseau par défaut : 255.255.0.0 Réseau de classe C -> Masque de sous-réseau par défaut : 255.255.255.0
4	Saisissez l'adresse IP de la **passerelle**. (Voir les *remarques 1 et 4*.)	Sur le réseau LAN, la passerelle doit se trouver sur le même segment que TwidoPort. En règle générale, cette information vous est fournie par votre administrateur réseau. Notez que l'application ne fournit aucune valeur par défaut ; vous devez renseigner ce champ avec une adresse de passerelle valide.
5	Validez la configuration et transférez-la vers l'automate Twido.	
6	Mettez l'automate Twido hors tension, puis remettez-le sous tension.	
Remarque 1	**Consultez votre administrateur réseau ou système pour obtenir les paramètres IP valides du réseau.**	
Remarque 2	**Pour une bonne communication sur le réseau, les équipements connectés doivent avoir une adresse IP unique. Lorsque TwidoPort est connecté au réseau, il recherche la présence de doublons d'adresse IP. Si un doublon est détecté sur le réseau, le voyant d'état clignote 4 fois à intervalles réguliers. Vous devez alors renseigner ce champ avec une adresse IP unique.**	
Remarque 3	**Utilisez le masque de sous-réseau par défaut, à moins que le module TwidoPort ne comporte des exigences particulières en matière de sous-réseau.**	
Remarque 4	**S'il n'existe aucune passerelle sur le réseau, saisissez simplement l'adresse IP de TwidoPort dans le champ Adresse de la passerelle.**	

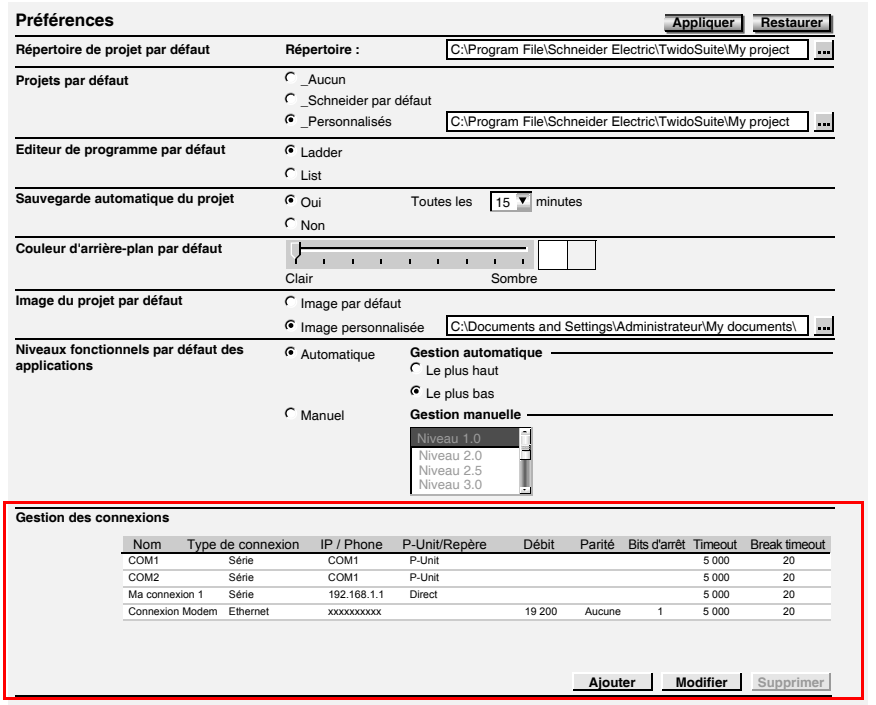
Configuration d'une connexion Ethernet dans TwidoSuite

Pour permettre au PC exécutant TwidoSuite et à l'automate Twido de communiquer sur le réseau Ethernet.

Sélectionnez **Préférences**.

Résultat :

La boîte de dialogue suivante **Gestion des connexions** apparaît :



Etape	Action
1	Cliquez sur le bouton **Ajouter** dans la boîte de dialogue **Gestion des connexions**. **Résultat :** Une ligne de connexion supplémentaire est ajoutée. Elle comprend les paramètres de connexion par défaut conseillés. Vous devez modifier ces paramètres. **Remarque :** Pour définir une nouvelle valeur dans un champ, vous avez deux possibilités : ● Sélectionnez le champ souhaité, puis cliquez sur le bouton **Modifier**. ● Cliquez deux fois sur le champ souhaité.
2	Dans le champ **Nom**, saisissez un nom descriptif pour la nouvelle connexion. Un nom valide contient au maximum 32 caractères alphanumériques.

Etape	Action
3	Cliquez dans le champ **Type de connexion** pour dérouler la liste qui inclut les éléments suivants : Série, Ethernet et USB (le cas échéant). Sélectionnez **Ethernet** puisque vous configurez une nouvelle connexion Ethernet entre un PC et un automate Twido prenant en charge Ethernet.
4	Dans le champ **IP / Phone**, saisissez une **adresse IP** qui correspond aux informations IP de l'automate Twido auquel vous souhaitez vous connecter. **Adresse IP :** Saisissez l'adresse IP statique précédemment spécifiée pour votre automate Twido.
5	Vous pouvez renseigner le champ **P-Unit/Repère** une fois que vous avez sélectionné le champ IP / Phone. Pour une **connexion Ethernet**, la valeur par défaut de P-Unit/Repère est Direct. Cette **valeur par défaut doit être changée pour la valeur 1** (ou pour l'adresse utilisée précédemment pour l'automate). Pour une connexion de type série, la valeur par défaut est **P-Unit**. Lorsqu'un de ces champs est sélectionné, les trois champs suivants (Débit, Parité et Bits d'arrêt) sont désactivés. Si vous ne connaissez pas l'adresse de l'automate, @ vous permet de la sélectionner plus tard, une fois que le programme a été transféré. *(Une fenêtre s'affiche avant la première connexion pour vous permettre de choisir l'automate vers lequel vous allez effectuer un transfert, avec une plage comprise entre 1 et 247, 1 étant la valeur d'adresse par défaut.)*
6	Utilisez les paramètres par défaut dans les champs **Timeout** et **Break timeout**, à moins que vous n'ayez des besoins spécifiques. (Pour plus d'informations, reportez-vous à la section (Gestion des connexions Ethernet, Twido, Guide de programmation).)
7	Cliquez sur **OK** pour enregistrer les nouveaux paramètres de connexion et fermer la boîte de dialogue Gestion des connexions. **Résultat :** Toutes les nouvelles connexions sont ajoutées à la table **Préférences** → **Gestion des connexions** et à la table **Programme** → **Mise au point** → **Connecter** → **Sélectionner une connexion**.

Configuration BootP

Processus BootP

TwidoPort attend une réponse du serveur BootP dans les deux minutes suivant la transmission de sa requête BootP. En cas d'échec, TwidoPort utilise la configuration IP par défaut, élaborée à partir d'une adresse MAC de ce type :

85	**16**	**MAC[4]**	**MAC[5]**

Adresse MAC

La structure de l'adresse MAC se présente comme suit :

`MAC[0] MAC[1] MAC[2] MAC[3] MAC[4] MAC[5].`

Par exemple, si l'adresse MAC est `0080F4012C71`, l'adresse IP par défaut est `85.16.44.113`.

12.2 Configuration Telnet de TwidoPort

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre explique la procédure de configuration du module TwidoPort ConneXium avec une session Telnet.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Présentation de la configuration Telnet	349
Menu principal Telnet	350
Paramètres IP/Ethernet	351
Configuration des paramètres série	353
Configuration de la passerelle	354
Configuration de la sécurité	355
Statistiques Ethernet	356
Serial Statistics	357
Enregistrement de la configuration	358
Restauration des paramètres par défaut	359
Mise à niveau du microprogramme TwidoPort	360
Vous avez oublié votre mot de passe et/ou votre configuration IP ?	362

Présentation de la configuration Telnet

Vue d'ensemble de la configuration Telnet

Configurez le module TwidoPort au cours d'une session Telnet (à l'aide d'un client Telnet compatible VT100) lorsqu'une configuration Twido spécifique est introuvable ou lorsqu'une requête BootP ne reçoit pas de réponse au bout de deux minutes (entraînant l'utilisation de l'adresse IP par défaut).

Conditions préalables à la configuration Telnet

Note : Exigences liées à Telnet TwidoPort
Lorsque vous configurez TwidoPort avec Telnet, assurez-vous que :
- TwidoPort est alimenté (à partir d'un automate Twido) via sa connexion série.
- `L'écho local` de Telnet est **`désactivé`**.

Pour pouvoir utiliser Telnet, ajoutez l'adresse IP par défaut de TwidoPort (ou celle configurée) à la table de routage de votre ordinateur en utilisant la commande:

```
C:\> route add 85.0.0.0 mask 255.0.0.0 local_IP_address_of_PC
```

Exemple :

Si l'adresse IP de votre ordinateur est `192.168.10.30` et que l'adresse IP par défaut (ou configurée) du module TwidoPort est `85.16.44.113`, la commande appropriée est la suivante :

```
C:\> route add 85.0.0.0 mask 255.0.0.0 192.168.10.30
```

Menu principal Telnet

Ouverture du menu principal Telnet

Lorsque vous ouvrez une session Telnet (en tapant par exemple `telnet 85.16.44.113` à l'invite ou en utilisant le programme Hyperterminal™ de Windows™), le menu principal Telnet apparaît après avoir appuyé sur **Entrée** :

```
        Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                (c) 2004 Schneider Automation Inc

1) IP/Ethernet Settings
                    IP Source: DEFAULT
                IP Address: 85.16.44.113
            Default Gateway: 85.16.44.113
                    Netmask: 0.0.0.0
        Ethernet Frame Type: ETHERNETII

2) Serial Configuration
        Baud Rate: 19200
        Data Bits: 8
           Parity:  NONE
        Stop Bits: 1
         Protocol:  RTU

3) Gateway Configuration
          Slave Address Source: UNIT_ID
              Gateway Mode: SLAVE
                MB Broadcasts: ENABLED

4) Security Configuration

5) Ethernet Statistics

6) Serial Statistics

Commands: D)efault settings, S)ave, F)irmware Upgrade, Q)uit without save
     Select Command or Parameter(1..6) to change:
```

Paramètres IP/Ethernet

Configuration des paramètres IP/Ethernet

Pour modifier les paramètres IP/Ethernet, procédez comme suit :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **1** pour modifier la source IP sur **STORED**, puis appuyez sur ***Entrée.***	Il se peut que la source IP soit déjà définie sur **STORED**.
3	Définissez les paramètres IP appropriés manuellement. (Voir *Paramètres Ethernet TwidoPort* après ce tableau.)	Exemples de paramètres complémentaires : • Adresse IP • Passerelle par défaut • Masque réseau • Type de trame Ethernet
4	Sélectionnez **R**, puis appuyez sur ***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)

IP Source

L'option **Source IP** sélectionnée indique l'emplacement à partir duquel la configuration IP est obtenue :

- STORED : mémoire flash locale.
- SERVED : serveur BootP.
- TWIDO : automate Twido.

L'adresse IP par défaut (DEFAULT) est dérivée de l'adresse MAC. (Par définition, la valeur par défaut ne peut pas être sélectionnée.)

Note : Une configuration IP correcte dans l'automate Twido annule la sélection de l'utilisateur.

Exemples de paramètres Ethernet

L'illustration suivante est un exemple des paramètres Ethernet TwidoPort :

```
Telemecanique 499 TWD 01 100 Configuration and Diagnost
        (c) 2004 Schneider Automation Inc

IP/Ethernet Settings
-------------------------------------------------
        1)IP Source: DEFAULT
        2)IP Address: 85.16.44.113
        3)Default Gateway: 85.16.44.113
        4)Netmask: 0.0.0.0
        5)Ethernet Frame Type: ETHERNET2
-------------------------------------------------
Commands: R)eturn to Main Menu
Select Command or Parameter(1..N) to change:
```

Configuration des paramètres série

Avant-propos

Note : Dans des conditions d'utilisation normales, vous n'êtes pas obligé de configurer les paramètres série de TwidoPort, car ce module prend en charge un algorithme de détection automatique (autobaud) rendant toute configuration série inutile.

Configuration des paramètres série

Pour configurer les paramètres série TwidoPort

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **2** pour modifier les paramètres série.	Voir l'illustration ci-dessous.
3	Vérifiez ou réinitialisez les paramètres.	Exemples de paramètres complémentaires : • Débit • Bits de données • Parité • Bits d'arrêt • Protocole
4	Sélectionnez **R**, puis appuyez sur ***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)

Exemple de paramètres série

L'illustration suivante est un exemple des paramètres série TwidoPort :

```
Telemecanique 499 TWD 01 100 Configuration and Diagnostics
        (c) 2004 Schneider Automation Inc

 Serial Configuration
        1) Baud Rate:  19200
        2) Data Bits:  8
        3)    Parity:  NONE
        4) Stop Bits:  1
            Protocol:  RTU

Commands: R)eturn to Main Menu
Select Command or Parameter(1..N) to change:
```

Configuration de la passerelle

Avant-propos

Note : Généralement, vous n'êtes pas obligé de configurer les paramètres de la passerelle TwidoPort.

Configuration des paramètres de la passerelle

Pour configurer la passerelle TwidoPort

Etape	Action		
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).	
2	Sélectionnez (saisissez) **3** pour modifier les paramètres de la passerelle.	Voir l'illustration ci-dessous.	
3	Les paramètres de passerelle suivants sont disponibles :		
	(1) Source de l'adresse esclave	FIXED	Si la source de l'adresse esclave est définie sur **FIXED**, utilisez l'adresse Modbus de l'automate Twido. Les adresses valides sont comprises entre 1 et 247.
		UNIT_ID	L'ID d'unité de la trame Modbus/TCP est utilisé.
	(2) Mode passerelle	SLAVE	Seule option disponible pour cette version.
	(3) Diffusions MB	DISABLED	Aucun message à diffusion générale n'est envoyé sur le port série TwidoPort.
		ENABLED	Des messages à diffusion générale sont envoyés depuis le port série de l'automate TwidoPort. *(Voir la section Remarque ci-dessous.)*
4	Sélectionnez **R**, puis appuyez sur ***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)	
Remarque	**Twido ne prend pas en charge les messages Modbus à diffusion générale.**		

Exemple de paramètres de la passerelle

L'illustration suivante est un exemple des paramètres de la passerelle TwidoPort :

```
Telemecanique 499 TWD 01 100 Configuration and Diagnostics
        (c) 2004 Schneider Automation Inc

Gateway Configuration
        1) Slave Address Source: UNIT_ID
        2) Slave Address: 20
        3) Gateway Mode: SLAVE
        4) MB Broadcasts: ENABLED

Commands: R)eturn to Main Menu
Select Command or Parameter(1..4) to change: _
```

Configuration de la sécurité

Configuration des paramètres de sécurité

Pour modifier le mot de passe par défaut, procédez comme suit :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **4**, puis appuyez sur ***Entrée.***	L'écran Configuration de la sécurité apparaît.
3	Sélectionnez **C**, puis appuyez sur ***Entrée.***	
4	Saisissez l'ancien mot de passe.	Les utilisateurs autorisés connaissent déjà ce mot de passe (par défaut, **USERUSER**).
5	Entrez le nouveau mot de passe.	Entrez de nouveau le nouveau mot de passe. *(Voir la remarque ci-dessous.)*
6	Entrez de nouveau le nouveau mot de passe.	Lisez la remarque ci-dessous concernant les mots de passe acceptés.
7	Sélectionnez **R**, puis appuyez sur***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)
Remarque	**Informations sur le mot de passe :** ● **longueur minimale : 4 caractères** ● **longueur maximale : 10 caractères** ● **caractères autorisés : de 0 à 9, de a à z, de A à Z (pas d'espace)**	

Statistiques Ethernet

Affichage des statistiques Ethernet

Pour afficher les statistiques Ethernet du module TwidoPort :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **5** pour afficher l'écran **Ethernet Module Statistics**.	Reportez-vous à l'illustration figurant sous ce tableau.
3	Appuyez sur **Entrée** pour rafraîchir l'écran.	
4	Appuyez sur **C** pour effacer les statistiques, puis appuyez sur ***Entrée***.	Tous les compteurs sont remis à zéro.
5	Sélectionnez **R**, puis appuyez sur ***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)

Ecran Ethernet Module Statistics

Ecran **Ethernet Module Statistics** de TwidoPort :

```
                Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                        (c) 2004 Schneider Automation Inc
                        ETHERNET MODULE STATISTICS
-------------------------------------------------------------------------------
Status: 0x9103                          IP Address: 192.168.1.141
System Log Entry: No                    Mac Address: 0:80:f4:0:4c:18
Transmit Speed: 100BASE-T               Subnet Mask: 255.255.0.0
Full/Half Duplex: Half Duplex           Gateway Address: 192.168.1.1
-------------------------------------------------------------------------------
Transmit Statistics     Receive Statistics      Functioning Errors

Transmits: 63           Receives: 532            Missed Packets: 0
Transmit Retries: 0     Framing Errors: 0        Collision Errors: 0
Lost Carrier: 0         Overflow Errors: 0       Transmit Timeouts: 0
Late Collision: 0       CRC Errors: 0            Memory Errors: 0
Tx Buffer Errors: 0     Rx Buffer Errors: 0      Net Interface Restarts: 0
SILO Underflow: 0
-------------------------------------------------------------------------------
Broadcast Packets Received: 37          Multicast Packets Received: 7
-------------------------------------------------------------------------------
Commands: [Enter] to Refresh, C)lear Statistics, R)eturn to Main Menu
```

Serial Statistics

Affichage de l'écran Serial Statistics

Pour afficher les statistiques série du module TwidoPort :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **6** pour afficher l'écran **Serial Statistics**, puis appuyez sur **Entrée**.	Reportez-vous à l'illustration figurant sous ce tableau. Les statistiques de l'interface série sont mises à jour.
3	Appuyez sur **C** pour effacer les statistiques, puis appuyez sur ***Entrée***.	Tous les compteurs sont remis à zéro.
4	Sélectionnez **R**, puis appuyez sur ***Entrée.***	Le menu principal Telnet apparaît. (Il est peut-être nécessaire d'appuyer de nouveau sur ***Entrée*** pour mettre à jour l'écran.)

Ecran Serial Statistics

Ecran **Serial Statistics** du module TwidoPort :

```
                Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                        (c) 2004 Schneider Automation Inc

---------------------- SERIAL STATISTICS --------------------------------

Serial Bus Statistics
        Bus Message Count: 8284
        Bus Comm. Error Count: 0
Modbus Slave Statistics
        Slave Message Count: 4142
        Slave Exception Error Count: 3187
        Slave No Response Count: 0

-------------------------------------------------------------------------

Commands: [Enter] to Refresh, C)lear Statistics, R)eturn to Main Menu
```

Enregistrement de la configuration

Enregistrement de la configuration

Pour enregistrer les modifications apportées à votre configuration, entrez le mot de passe de la configuration :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez **S**, puis appuyez sur ***Entrée.***	
3	Entrez le mot de passe de la configuration.	Le mot de passe par défaut est **USERUSER**. *(Voir la section Remarque ci-dessous.)*
Remarque	**Pour obtenir davantage d'informations sur la définition d'un mot de passe de sécurité personnalisé, reportez-vous à la *p. 355*.**	

Ecran de confirmation Save Configuration

Ecran de confirmation **Save Configuration** du module TwidoPort :

```
                Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                        (c) 2004 Schneider Automation Inc
                        SAVE CONFIGURATION
--------------------------------------------------------------------------------

        Configuration successfully stored to Twido.
        Reboot your module for the new Configuration to be in effect.

        Rebooting in 5 Seconds. You will lose your telnet connection.

Connection to host lost.
```

Restauration des paramètres par défaut

Restauration des paramètres par défaut

Pour restaurer les paramètres par défaut du module TwidoPort :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez **D** pour afficher l'écran **Default Configuration**.	Reportez-vous à l'illustration figurant sous ce tableau.
3	Appuyez sur **Entrée.**	Appuyez sur **Entrée** pour afficher le menu principal.
4	Enregistrez la configuration par défaut.	Voir la section Enregistrement de la configuration (voir *p. 358*) précédente.

L'écran Default Configuration

Ecran **Default Configuration** du module TwidoPort :

```
        Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                (c) 2004 Schneider Automation Inc
                DEFAULT CONFIGURATION
------------------------------------------------------------------------


IP Address: 192.168.2.102
Gateway Address: 192.168.2.102
Subnet Mask: 255.255.0.0
Frame Type: Ethernet II

Serial Mode: 19200-8-N-1

Gateway Mode: Modbus/RTU Slave Attached
              Broadcasts Disabled, Slave Address Source=Unit ID

Configuration Password: USERUSER

You must (S)ave the configuration to make it active.

Returning to Main Menu in 2 Seconds, Hit Enter to refresh._
```

Mise à niveau du microprogramme TwidoPort

Avant-propos

Note :
1. Procurez-vous une version plus récente du microprogramme TwidoPort avant d'essayer de le mettre à niveau à l'aide de ces instructions.
2. Arrêtez le processus avant de mettre à niveau le microprogramme.
3. La communication Modbus ne sera pas disponible pendant la mise à niveau du microprogramme.

Mise à niveau du microprogramme

Pour mettre à niveau le microprogramme TwidoPort actuel avec la dernière version que vous avez obtenue, procédez comme suit :

Etape	Action	Commentaire
1	Démarrez une session Telnet.	Suivez les instructions fournies pour ouvrir le menu principal Telnet (voir *p. 350*).
2	Sélectionnez (saisissez) **`F`** pour lancer la mise à niveau du microprogramme.	Cinq secondes après avoir sélectionné **`F`** (mise à niveau du microprogramme), le module TwidoPort est réinitialisé et vous perdez la connexion Telnet.
3	Sur la ligne de commande, saisissez : **`ftp`** et l'adresse IP du module TwidoPort.	Par exemple : **`ftp 85.16.44.113`**
4	Entrez : **`ftptwd`**	A l'invite du nom de connexion.
5	Entrez : **`cd fw`**	Le répertoire **`fw`** s'affiche.
6	Entrez : **`put App.out`**. *(Voir les remarques 1 et 2.)*	Un message indiquant que la commande ftp a réussi s'affiche. *(Voir remarque 3.)*
Remarque 1	**Les noms de fichier prennent en compte les majuscules et les minuscules.**	
Remarque 2	**Assurez-vous que `App.out` se trouve dans le répertoire de travail courant du client ftp.**	
Remarque 3	**Un message indiquant que le module TwidoPort redémarrera automatiquement 5 secondes après une commande ftp réussie s'affiche.**	

Firmware Upgrade In-Progress

L'illustration suivante représente l'écran **Firmware Upgrade In-Progress** type :

```
Telemecanique 499 TWD 01 100 Configuration and Diagnostics
--------------------------------------------------

FIRMWARE UPGRADE IN-PROGRESS...
Module will reboot in 5 Seconds.
After Reboot, Connect via FTP to download new Firmware.

FTP Instructions:
        1) Connect via FTP:  ftp  192.168.2.160
        2) Change to /fw directory:  ftp>cd fw
        3) Download new fw:  ftp>put App.out

After the FTP download is complete, the module will reboot automatically
.

Rebooting now.  Goodbye.


Connection to host lost.
```

Mode noyau

En l'absence d'un microprogramme valide, le module TwidoPort passe en mode **noyau**. Si vous essayez d'utiliser Telnet pour vous connecter au module TwidoPort alors que le mode noyau est actif, les informations suivantes s'affichent à l'écran :

```
Telemecanique 499 TWD 01 100
------------------------------------------------
Kernel Version 90.02d

Download valid Exec,App.out, to leave kernel mode.

   To exit type 'quit' 'QUIT' or control D
```

Vous avez oublié votre mot de passe et/ou votre configuration IP ?

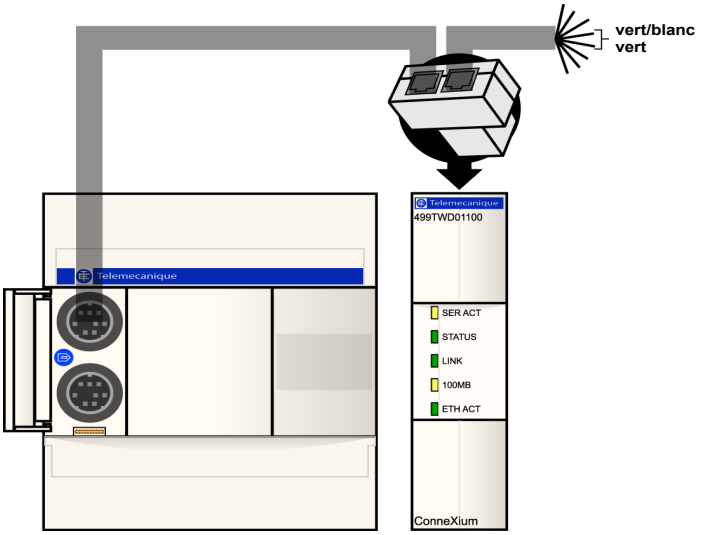
Connexion en mode de sauvegarde

Suivez les instructions suivantes pour vous connecter au module TwidoPort en mode de sauvegarde.

Etape	Action	Commentaire
1	Connectez la broche 3 à la broche 6 (mise à la terre) du connecteur série.	Schneider recommande l'utilisation d'un connecteur RJ45 T 170 XTS 04 100 *(reportez-vous à l'illustration ci-après).*
2	Connectez-vous au module TwidoPort via une connexion FTP *(voir la remarque).*	Le module TwidoPort utilise la configuration IP par défaut suivante : • Adresse IP : 192.168.2.102 • Masque de sous-réseau : 255.255.0.0 • Adresse de passerelle : 192.168.2.102 • Type de trame : Ethernet II
3	Localisez le fichier **fw/Conf.dat**.	Vous trouverez la configuration IP et le mot de passe dans le fichier **Conf.dat**.
4	Ouvrez le fichier **Conf.dat** dans un éditeur de texte.	
Remarque	**Aucun mot de passe n'est requis.**	

Connexion FTP

L'illustration suivante explique comment se connecter au module TwidoPort via une connexion FTP en mode de sauvegarde :



12.3 Fonctions de communication

Aperçu

Objet de cette section

Cette section présente les fonctions de communication prises en charge par la passerelle Ethernet du module TwidoPort ConneXium.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonctionnalités Ethernet	364
Protocole de communication Modbus/TCP	365
Codes de fonction Modbus pris en charge localement	366

Fonctionnalités Ethernet

Introduction

Le module TwidoPort ConneXium ajoute une connexion Ethernet à la gamme de produits Twido de Télémécanique. Il s'agit de la passerelle entre un équipement Modbus/RTU (RS485) Twido et la couche physique des réseaux Modbus/TCP en mode esclave. Le module TwidoPort ne requiert pas d'alimentation distincte, car il est alimenté via le port série de l'automate Twido. Ce module passerelle prend en charge le mode esclave uniquement.

Fonctionnalités Ethernet

TwidoPort prend en charge les fonctions Ethernet suivantes :

- **Autonégociation**
  TwidoPort prend en charge l'autonégociation 10/100TX. Il ne communique qu'en mode semi-duplex.
- **Auto-MDI/MDI-X**
  TwidoPort prend en charge la commutation automatique des paires de câbles de transmission et de réception pour établir la communication avec l'équipement final (auto-MDI/MDI-X). Il établit donc une interconnexion claire entre l'infrastructure ou les terminaux et les câbles croisés ou droits.

Protocole de communication Modbus/TCP

A propos de Modbus

Le protocole Modbus est un protocole maître/esclave permettant à un maître de demander des réponses auprès des esclaves ou d'agir en fonction de leurs demandes. Le maître peut s'adresser à chaque esclave ou envoyer un message de diffusion générale à l'ensemble des esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les réponses aux requêtes de diffusion générale du maître ne sont pas renvoyées.

A propos des communications Modbus/TCP

TwidoPort prend en charge jusqu'à 8 connexions Modbus/TCP simultanées. Si vous tentez d'utiliser plus de 8 connexions, cela réduit les performances, car TwidoPort ferme la connexion la plus ancienne afin de prendre en charge la nouvelle connexion.

Fonctionnement

Les clients Modbus/TCP peuvent communiquer avec Twido via TwidoPort, qui constitue un pont entre les équipements Twido (Modbus/RTU sur une liaison série RS485) et Modbus/TCP sur les réseaux Ethernet.

Note : Lors de la mise en œuvre de TwidoPort sur un réseau, la configuration système requise doit être adaptée à la bande passante limitée inhérente associée aux connexions en série. Les performances maximales devraient atteindre environ 40 transactions Modbus par seconde. Il est plus efficace de demander plusieurs registres dans une seule requête que d'effectuer des requêtes séparées pour chaque registre.

Vous ne pouvez pas initier les requêtes en lecture ou en écriture à partir de l'automate Twido via TwidoPort.

Codes de fonction Modbus pris en charge localement

Liste des codes de fonction

TwidoPort répond aux codes de fonction Modbus ci-dessous, pris en charge localement, uniquement lorsque l'ID d'unité est paramétré sur 254. (Les codes de fonction pris en charge localement sont ceux qui obtiennent une réponse directement de TwidoPort et non de l'automate Twido.)

Code de fonction Modbus	Code de sous-fonction	OPCODE	Description
8	0	N/A	Renvoie les données de requête.
8	10	N/A	Remet les compteurs à zéro.
8	11	N/A	Renvoie le nombre de messages du bus.
8	12	N/A	Renvoie le nombre d'erreurs de communication du bus.
8	13	N/A	Renvoie le nombre d'erreurs d'exception du bus.
8	14	N/A	Renvoie le nombre de messages de l'esclave.
8	15	N/A	Renvoie le nombre de messages sans réponse de l'esclave.
8	21	3	Fournit les statistiques Ethernet.
8	21	4	Efface les statistiques Ethernet.
43	14	N/A	Lit l'ID de l'équipement (voir *remarque 1*.)
Remarque 1	**TwidoPort ne prend en charge que les ID d'objet de base du code de fonction permettant de lire l'identifiant de l'équipement, avec un accès individuel ou en continu.**		

Note : Voir les spécifications Modbus relatives au format des messages et aux classes d'accès, à l'adresse *www.modbus.org*.

Fonctionnement de l'afficheur

13

Présentation

Objet de ce chapitre

Ce chapitre offre des informations sur l'utilisation de l'afficheur optionnel Twido.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Afficheur	368
Informations d'identification et états de l'automate	371
Variables et objets système	373
Paramètres de port série	379
Horloge calendaire	380
Facteur de correction de l'horodateur	381

Afficheur

Introduction

L'afficheur est une option de Twido qui permet d'afficher et de contrôler les données de l'application et quelques fonctions de l'automate, telles que l'état de fonctionnement et l'horodateur (RTC). Cette option est disponible sous la forme d'une cartouche (TWDXCPODC) pour les automates compacts ou d'un module d'expansion (TWDXCPODM) pour les automates modulaires.

L'afficheur dispose de deux modes de marche :

- Mode affichage : affiche simplement les données.
- Mode édition : permet de modifier les données.

Note : L'afficheur est mis à jour selon un intervalle défini dans le cycle de scrutation de l'automate. Cela peut provoquer des erreurs d'interprétation de l'affichage des sorties dédiées pour les impulsions %PLS et %PWM. Au moment où ces sorties sont échantillonnées, leur valeur est toujours égale à zéro et cette valeur est affichée.

Ecrans et fonctions

L'afficheur propose différents écrans à partir desquels vous pouvez accéder aux fonctions associées.

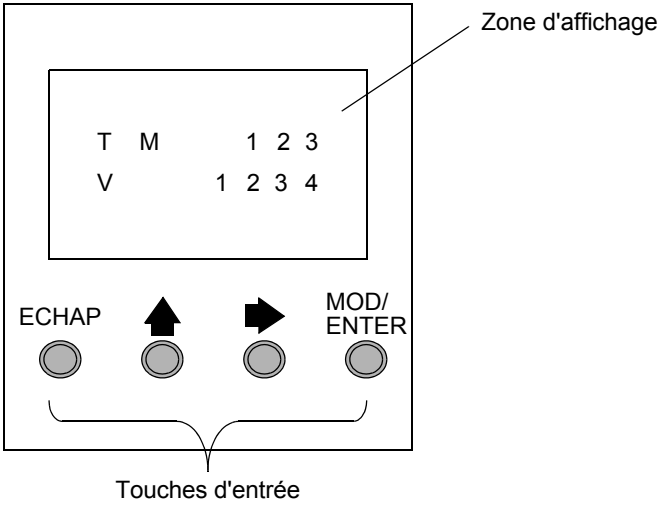
- Informations sur l'identification et l'état de l'automate : Afficheur
  Affiche la révision du microprogramme et l'état de l'automate. Modifie l'état de l'automate à l'aide des commandes d'exécution, d'initialisation et d'arrêt.
- Variables et objets système : écran Données
  Sélection des données de l'application par le repère : %I, %Q et tous les autres objets logiciels de la base automate. Contrôle et modification de la valeur de l'objet donnée logicielle sélectionné.
- Paramètres du port série : écran Communications
  Affichage et modification des paramètres du port de communication.
- Horloge calendaire : écran Date/Heure
  Affichage et configuration de la date et de l'heure courantes (lorsque l'horodateur est installé).
- Correction de l'horodateur : facteur RTC
  Affichage et modification de la valeur de correction de l'horodateur en option.

Note :

1. Les automates compacts de la série TWDLC••40DRF et les automates Twido Extreme TWDLEDCK1 disposent d'un horodateur intégré.
2. Pour tous les autres automates, l'horloge calendaire et la correction RTC ne sont disponibles que lorsque la cartouche horodateur en option (TWDXCPRTC) est installée.

Illustration

L'illustration suivante présente une vue de l'afficheur. Il est composé d'une zone d'affichage (ici en mode normal) et de quatre touches d'entrée.



Zone d'affichage

L'afficheur est composé d'un écran à cristaux liquides pouvant afficher jusqu'à deux lignes de caractères.

- La première ligne de l'écran est composée de trois caractères de 13 segments et de quatre caractères de 7 segments.
- La seconde ligne est composée d'un caractère de 13 segments, d'un caractère de 3 segments (pour les signes plus et moins) et de cinq caractères de 7 segments.

Note : En mode normal, la première ligne indique un nom d'objet et la deuxième ligne affiche sa valeur. En mode de données, la première ligne affiche la valeur %SW68 et la deuxième ligne la valeur %SW69.

Touches d'entrée

Les fonctions des quatre touches d'entrée dépendent du mode de l'afficheur.

Touche	En mode affichage	En mode édition
ESC		Annulation des modifications et retour à l'écran précédent.
▲		Accès à la valeur suivante d'un objet en cours de modification.
➡	Passage à l'écran suivant.	Accès au type d'objet suivant à modifier.
MOD/ENTER	Passage en mode édition.	Validation des modifications et retour à l'écran précédent.

Sélection et navigation entre les écrans

L'affichage ou l'écran initial de l'afficheur présente des informations sur l'identification et l'état de l'automate. Appuyez sur la touche ➡ pour passer d'un affichage à l'autre. Les écrans de l'horloge calendaire ou le facteur de correction RTC apparaissent uniquement lorsque la cartouche horodateur en option (TWDXCPRTC) est détectée sur l'automate.

Appuyez sur la touche ECHAP pour revenir à l'écran initial. Dans la plupart des écrans, le fait d'appuyer sur la touche ECHAP permet de revenir à l'écran relatif aux informations d'identification et d'état de l'automate. Le fait d'appuyer sur la touche ECHAP permet de revenir à la saisie du premier objet système ou de l'objet système initial uniquement lors de la modification de variables et d'objets système autres que l'entrée initiale (%I0.0.0).

Pour modifier la valeur d'un objet, appuyez à nouveau sur la touche MOD/ENTER au lieu d'appuyer sur la touche ➡ pour accéder au premier chiffre de la valeur.

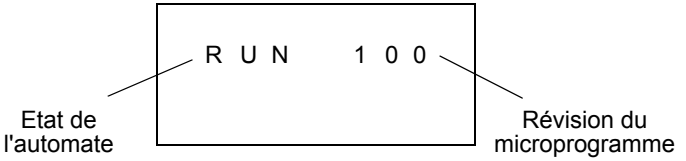
Informations d'identification et états de l'automate

Introduction

L'écran initial de l'afficheur optionnel Twido présente des informations sur l'identification et sur l'état de l'automate.

Exemple

Comme l'illustre le schéma suivant, la version du microprogramme est affichée dans le coin supérieur droit de la zone d'affichage, l'état de l'automate dans le coin supérieur gauche.



Etats de l'automate

L'automate peut se trouver dans l'un des états suivants :

- **NCF : Non configuré**
  L'automate demeure en état NCF jusqu'à ce qu'une application soit chargée. Aucun autre état n'est permis avant le chargement du programme de l'application. Vous pouvez tester les E/S en modifiant le bit système S8 (reportez-vous à la rubrique *p. 674*).
- **STP : Stopped**
  Dès qu'une application est chargée sur l'automate, ce dernier passe à l'état STP. Dans cet état, l'application ne fonctionne pas. Les entrées sont mises à jour et les valeurs des données restent inchangées. Les sorties ne sont pas mises à jour lorsque l'automate est arrêté.
- **INI : Initial**
  Seul un automate se trouvant à l'état STP peut passer à l'état INI. L'application n'est pas en cours d'exécution. Les entrées de l'automate sont mises à jour et les valeurs des données sont initialisées. Les sorties ne sont pas mises à jour lorsque l'automate est en cours d'initialisation.
- **RUN : En cours d'exécution**
  Dans cet état, l'application fonctionne. Les entrées de l'automate sont mises à jour et les valeurs des données sont réglées par l'application. Il s'agit du seul état au cours duquel les sorties sont mises à jour.
- **HLT : Halt (Erreur d'application utilisateur)**
  L'exécution de l'application est arrêtée dès que l'automate passe à l'état ERR. Les entrées sont mises à jour et les valeurs des données restent inchangées. Dans cet état, les sorties ne sont pas mises à jour. Dans ce mode, le code de l'erreur est affiché dans la partie inférieure droite de l'afficheur. Ce code prend la forme d'une valeur décimale sans signe.
- **NEX : Not Executable (non exécutable)**
  Une modification en ligne a été apportée à la logique utilisateur. Conséquence : l'application n'est plus exécutable. Elle ne retrouvera cet état qu'une fois que toutes les causes de l'état Non Exec auront été résolves.

Affichage et modification des états de l'automate

L'afficheur vous permet de faire passer l'automate de l'état STP à l'état INI, de l'état STP à l'état RUN, ou de l'état RUN à l'état STP. Pour modifier l'état de l'automate, procédez comme suit :

Etape	Action
1	Appuyez sur la touche ▶ jusqu'à ce que l'écran Affichage des états de l'automate apparaisse (ou appuyez sur la touche ESC). L'état courant de l'automate apparaît dans le coin supérieur gauche de la zone d'affichage.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche ▲ pour sélectionner un état de l'automate.
4	Appuyez sur la touche MOD/ENTER pour accepter la valeur modifiée, ou sur la touche ESC pour ignorer les modifications apportées en mode édition.

Variables et objets système

Introduction

L'afficheur optionnel permet de contrôler et d'ajuster les données de l'application à l'aide des fonctionnalités suivantes :

- sélection des données de l'application par le repère (%I ou %Q, par exemple) ;
- contrôle de la valeur de l'objet/variable logiciel(le) sélectionné(e) ;
- modification de la valeur de l'objet donnée actuellement affiché (y compris le forçage des entrées et des sorties).

Variables et objets système

Le tableau suivant répertorie, dans leur ordre d'accès, les variables et objets système qui peuvent être affichés et modifiés via l'afficheur.

Objet	Variable/Attribut	Description	Accès
Entrée	%Ix.y.z	Valeur	Lecture/Forçage
Sortie	%Qx.y.z	Valeur	Lecture/Ecriture/ Forçage
Temporisateur	%TMX.V %TMX.P %TMX.Q	Valeur courante Valeur de présélection Terminé	Lecture/Ecriture Lecture/Ecriture Lecture
Compteur	%Cx.V %Cx.P %Cx.D %Cx.E %Cx.F	Valeur courante Valeur de présélection Terminé Vide Plein	Lecture/Ecriture Lecture/Ecriture Lecture Lecture Lecture
Bit mémoire	%Mx	Valeur	Lecture/Ecriture
Mémoire mots	%MWx	Valeur	Lecture/Ecriture
Mot constant	%KWx	Valeur	Lecture
Bit système	%Sx	Valeur	Lecture/Ecriture
Mot système	%SWx	Valeur	Lecture/Ecriture
Entrée analogique	%IWx.y.z	Valeur	Lecture
Sortie analogique	%QWx.y.z	Valeur	Lecture/Ecriture
Compteur rapide (FC)	%FCx.V %FCx.VD* %FCx.P %FCx.PD* %FCx.D	Valeur courante Valeur courante Valeur de présélection Valeur de présélection Terminé	Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture

Objet	Variable/Attribut	Description	Accès
Compteur très rapide (VFC)	%VFCx.V %VFCx.VD* %VFCx.P %VFCx.PD* %VFCx.U %VFCx.C %VFCx.CD* %VFCx.S0 %VFCx.S0D* %VFCx.S1 %VFCx.S1D* %VFCx.F %VFCx.T %VFCx.R %VFCx.S	Valeur courante Valeur courante Valeur de présélection Valeur de présélection Sens de comptage Valeur de capture Valeur de capture Valeur seuil 0 Valeur seuil 0 Valeur seuil 1 Valeur seuil 1 Débordement Base temps Activation sortie réflexe Activation entrée réflexe	Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture
Mot entrée réseau	%INWx.z	Valeur	Lecture
Mot sortie réseau	%QNWx.z	Valeur	Lecture/Ecriture
Grafcet	%Xx	Bit pas	Lecture
Générateur d'impulsions	%PLS.N %PLS.ND* %PLS.P %PLS.D %PLS.Q	Nombre d'impulsions Nombre d'impulsions Valeur de présélection Terminé Sortie en courant	Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture Lecture
Modulateur de largeur d'impulsion	%PWM.R %PWM.P	Rapport Valeur de présélection	Lecture/Ecriture Lecture/Ecriture
Programmateur cyclique	%DRx.S %DRx.F	Numéro du pas courant Plein	Lecture Lecture
Fonction pas à pas	%SCx.n	Bit de fonction pas à pas	Lecture/Ecriture
Registre	%Rx.I %Rx.O %Rx.E %Rx.F	Entrée Sortie Vide Plein	Lecture/Ecriture Lecture/Ecriture Lecture Lecture
Registre bits à décalage	%SBR.x.yy	Bit de registre	Lecture/Ecriture
Message	%MSGx.D %MSGx.E	Terminé Erreur	Lecture Lecture
Entrée esclave AS-Interface	%IAx.y.z	Valeur	Lecture/Forçage
Entrée analogique esclave AS-Interface	%IWAx.y.z	Valeur	Lecture
Sortie esclave AS-Interface	%QAx.y.z	Valeur	Lecture/Ecriture/ Forçage
Sortie analogique esclave AS-Interface	%QWAx.y.z	Valeur	Lecture/Ecriture
Entrée PDO d'un esclave CANopen	%IWCx.y.z	Valeur de mot simple	Lecture
Sortie PDO d'un esclave CANopen	%QWCx.y.z	Valeur de mot simple	Lecture/Ecriture

Remarques :

1. (*) correspond à une variable de mot double 32 bits. L'option de double mot est disponible sur tous les automates à l'exception des automates Twido TWDLC•A10DRF.
2. Etant donné que Twido utilise l'affectation de mémoire dynamique, les variables n'apparaîtront pas si elles ne sont pas utilisées dans une application.
3. Si la valeur de %MW est supérieure à +32 767 ou inférieure à -32 768, l'afficheur continue de clignoter.
4. Si la valeur de %SW est supérieure à 65 535, l'afficheur continue de clignoter, sauf pour %SW0 et %SW11. Si la valeur saisie est supérieure à la limite, elle est remplacée par la valeur configurée.
5. Lorsqu'une valeur dépassant les limites est entrée pour %PLS.P, la valeur écrite est la valeur de saturation.

Affichage et modification des objets et des variables

Vous pouvez accéder à chaque type d'objet système en commençant par l'objet entrée (%I), en progressant de façon séquentielle jusqu'à l'objet message (%MSG) et en revenant finalement à l'objet entrée (%I).

Pour afficher un objet système, procédez comme suit :

Etape	Action
1	Appuyez sur la touche ➡ jusqu'à ce que l'écran Affichage des données apparaisse. L'objet Entrée ("I") apparaît dans le coin supérieur gauche de la zone d'affichage. La lettre "I" (ou le nom de l'objet précédemment visualisé en donnée) ne clignote pas.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition. La lettre "I" de l'objet Entrée (ou le nom de l'objet précédemment visualisé en donnée) commence à clignoter.
3	Appuyez sur la touche ⬆ pour progresser de façon séquentielle dans la liste des objets.
4	Appuyez sur la touche ➡ pour progresser de façon séquentielle dans le champ d'un type d'objet et appuyez sur la touche ⬆ pour incrémenter la valeur de ce champ. Utilisez les touches ➡ et ⬆ pour consulter et modifier tous les champs de l'objet affiché.
5	Répétez les étapes 3 et 4 jusqu'à ce que l'édition soit terminée.
6	Appuyez sur la touche MOD/ENTER pour accepter les valeurs modifiées. Remarque : Le nom et le repère de l'objet doivent être validés pour pouvoir accepter ces modifications. Cela signifie qu'ils doivent exister dans la configuration de l'automate avant de pouvoir utiliser l'afficheur. Appuyez sur la touche ECHAP pour annuler les modifications apportées en mode édition.

Valeurs des données et formats d'affichage

En général, la valeur des données pour un objet ou une variable est affichée comme un entier avec signe ou sans signe dans la partie inférieure droite de la zone d'affichage. En outre, les zéros non significatifs sont supprimés de tous les champs pour l'affichage des valeurs. Le repère de chaque objet apparaît dans l'afficheur dans l'un des sept formats suivants :

- Format E/S
- Format E/S des esclaves AS-Interface
- Format E/S des esclaves CANopen
- Format bloc fonction
- Format simple
- Format E/S réseau
- Format fonction pas à pas
- Format registre bits à décalage

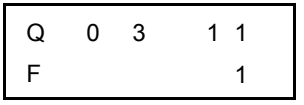
Format E/S

Les objets entrée/sortie (%I, %Q, %IW et %QW) présentent un repère en trois parties (ex. : %IX.Y.Z) et apparaissent sous la forme suivante :

- type d'objet et repère de l'automate dans la partie supérieure gauche ;
- repère de l'expansion dans la partie supérieure centrale ;
- voie d'E/S dans la partie supérieure droite.

Dans le cas d'une entrée (%I) et d'une sortie (%Q) simples, la lettre "U" pour un bit non forcé (unforced) ou la lettre "F" pour un bit forcé (forced) apparaît dans la partie inférieure gauche de l'écran. La valeur de forçage apparaît dans la partie inférieure droite de l'écran.

L'objet sortie %Q0.3.11 apparaît dans la zone d'affichage sous la forme suivante :



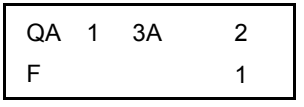
Format E/S des esclaves AS-Interface

Les objets E/S des esclaves AS-Interface (%IA, %QA, %IWA et %QWA) présentent un repère en quatre parties (ex. : %IAx.y.z) et apparaissent sous la forme suivante :

- type d'objet dans la partie supérieure gauche ;
- repère du maître AS-Interface sur le bus d'expansion dans la partie supérieure gauche ;
- repère de l'esclave sur le bus AS-Interface dans la partie supérieure droite ;
- voie d'E/S de l'esclave dans la partie supérieure droite.

Dans le cas d'une entrée (%IA) et d'une sortie (%QA) simples, la lettre "U" pour un bit non forcé (unforced) ou la lettre "F" pour un bit forcé (forced) apparaît dans la partie inférieure gauche de l'écran. La valeur de forçage apparaît dans la partie inférieure droite de l'écran.

L'objet sortie %QA1.3A.2 apparaît dans la zone d'affichage sous la forme suivante :

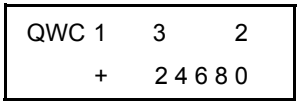


Format E/S des esclaves CANopen

Les objets E/S PDO des esclaves CANopen (%IWC et %QWC) disposent d'un repère en quatre parties (ex. : %IWCx.y.z) et apparaissent sous la forme suivante :

- type d'objet dans la partie supérieure gauche ;
- repère du maître CANopen sur le bus d'expansion dans la partie supérieure gauche ;
- repère de l'esclave sur le bus CANopen dans la partie supérieure droite ;
- voie d'E/S PDO de l'esclave dans le coin supérieur droit ;
- valeur avec signe de l'objet dans la partie inférieure.

Dans l'exemple ci-dessous, l'objet de sortie PDO %QWC1.3.2 contient la valeur signée +24 680 :



Format bloc fonction

Les blocs fonction (%TM, %C, %FC, %VFC, %PLS, %PWM, %DR, %R et %MSGj) disposent d'un repère en deux parties comprenant le numéro de l'objet et le nom d'une variable ou d'un attribut. Ils apparaissent sous la forme suivante :

- nom du bloc fonction dans la partie supérieure gauche ;
- numéro (ou instance) du bloc fonction dans la partie supérieure droite ;
- variable ou attribut dans la partie inférieure gauche ;
- valeur de l'attribut dans la partie inférieure droite.

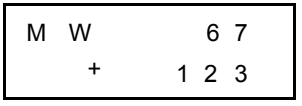
Dans l'exemple suivant, la valeur courante du temporisateur numéro 123 est réglée sur 1 234.

Format simple

Un format simple est utilisé pour les objets %M, %MW, %KW, %MD, %KD, %MF, %KF, %S, %SW et %X :

- numéro de l'objet dans la partie supérieure droite ;
- valeur avec signe pour les objets dans la partie inférieure.

Dans l'exemple suivant, le mot mémoire numéro 67 contient la valeur +123.

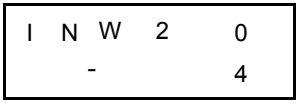


Format E/S réseau

Les objets E/S réseau (%INW et %QNW) apparaissent dans la zone d'affichage sous la forme suivante :

- nom de l'objet dans la partie supérieure gauche ;
- repère de l'automate dans la partie supérieure centrale ;
- numéro de l'objet dans la partie supérieure droite ;
- valeur avec signe de l'objet dans la partie inférieure.

Dans l'exemple suivant, le premier mot d'entrée réseau de l'automate distant configuré au repère distant n° 2 a pour valeur -4.

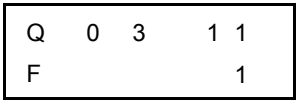


Format fonction pas à pas

Le format fonction pas à pas (%SC) affiche le numéro de l'objet et le bit de fonction pas à pas sous la forme suivante :

- nom et numéro de l'objet dans la partie supérieure gauche ;
- numéro du bit de fonction pas à pas dans la partie supérieure droite ;
- valeur du bit de fonction pas à pas dans la partie inférieure.

Dans l'exemple suivant, le bit numéro 129 de la fonction pas à pas numéro 3 est réglé sur 1.

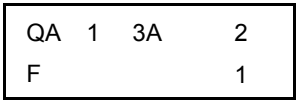


Format registre bits à décalage

Le registre bits à décalage (%SBR) apparaît dans la zone d'affichage sous la forme suivante :

- nom et numéro de l'objet dans la partie supérieure gauche ;
- numéro du bit de registre dans la partie supérieure droite ;
- valeur du bit de registre dans la partie inférieure droite.

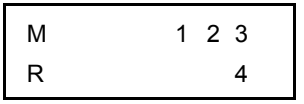
Vous trouverez ci-après un exemple de l'affichage du registre bit à décalage numéro 4.



Paramètres de port série

Introduction

L'afficheur vous permet de visualiser les paramètres du protocole et de modifier les repères de tous les ports série configurés à l'aide de TwidoSuite. Un maximum de deux ports série peut être utilisé. Dans l'exemple suivant, le premier port est configuré pour le protocole Modbus et porte le repère 123. Le second port est configuré en tant que liaison distante et porte le repère 4.



Affichage et modification des paramètres d'un port série

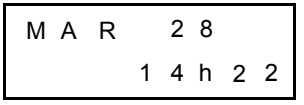
Les automates Twido peuvent gérer un maximum de deux ports série. Pour visualiser les paramètres des ports série sur l'afficheur :

Etape	Action
1	Appuyez sur la touche ➡ jusqu'à ce que l'écran Communications apparaisse. Une lettre, correspondant au paramètre de protocole du premier port (M, R ou A), est affichée dans le coin supérieur gauche de l'afficheur.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche ➡ jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche ⬆ pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que tous les paramètres de l'adresse aient été définis.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ESC pour les ignorer.
Remarque : Le repère fait partie des données de configuration de l'automate. Le changement de ses valeurs à l'aide de l'afficheur signifie que vous ne pouvez plus vous connecter à l'aide de TwidoSuite. TwidoSuite requiert un téléchargement pour être de nouveau à niveau.	

Horloge calendaire

Présentation

Les paramètres de date et d'heure ne peuvent être mis à jour depuis l'afficheur que si la cartouche optionnelle de l'horodateur (TWDXCPRTC) est installée sur votre automate Twido. Le mois apparaît dans la partie supérieure gauche de l'écran IHM. La valeur "RTC" figurera dans ce champ jusqu'à ce que des paramètres de date et d'heure valides aient été entrés. Le jour du mois apparaît dans la partie supérieure droite de l'écran. Cette heure est affichée au format dit "militaire". Les heures et les minutes sont affichées dans le coin inférieur droit de l'écran et sont séparées par la lettre "h". L'exemple suivant illustre ce qu'indiquerait l'afficheur, le 28 mars à 14:22.



Note :
1. Les automates compacts de la série TWDLCA•40DRF et les automates Twido Extreme TWDLEDCK1 disposent d'un RTC intégré.
2. Pour tous les autres automates, l'horloge calendaire et la correction RTC ne sont disponibles que lorsque la cartouche horodateur en option (TWDXCPRTC) est installée.

Affichage et modification de l'horloge calendaire

Pour afficher et modifier l'horloge calendaire, procédez comme suit :

Etape	Action
1	Appuyez sur la touche ▶ jusqu'à ce que l'écran de date et heure apparaisse. Le code du mois ("JAN" ou "FEV", par exemple) apparaît dans le coin supérieur gauche de la zone d'affichage. La mention "RTC" est affichée dans ce même coin tant que le mois n'a pas été défini.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche ▶ jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche ▲ pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que tous les paramètres de date et d'heure aient été définis.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ESC pour les ignorer.

Facteur de correction de l'horodateur

Présentation

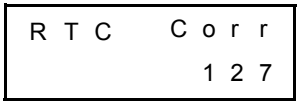
Pour chaque horodateur (RTC), une valeur de correction permet de corriger les imprécisions du cristal du module horodateur.

Vous pouvez afficher et modifier le facteur de correction de l'horodateur :
- à l'aide de l'afficheur ;
- en sélectionnant **Programme** → **Mise au point** → **Contrôler l'automate** → **Configurer horodateur** dans le panneau d'outils de tâches ;
- à l'aide de bits système et de mots système (reportez-vous à la section *p. 797*.

Facteur de correction de l'horodateur des bases compactes et modulaires

Pour les bases compactes et modulaires, le facteur de correction de l'horodateur est un nombre entier à 3 chiffres sans signe (de 0 à 127) qui s'affiche dans le coin inférieur droit de l'écran.

L'exemple suivant illustre un facteur de correction de 127.



Affichage et modification de la correction de l'horodateur

Pour afficher et modifier le facteur de correction de l'horodateur :

Etape	Action
1	Appuyez sur la touche ➡ jusqu'à ce que l'écran Affichage du facteur de correction RTC apparaisse. "RTC Corr" s'affiche dans la ligne supérieure de l'afficheur.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche ➡ jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche ⬆ pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que la valeur de correction RTC ait été définie.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ESC pour les ignorer.

Facteur de correction de l'horodateur des bases Twido Extreme

Pour les bases Twido Extreme TWDLEDCK1, le facteur de correction de l'horodateur doit être calculé comme suit :

Etape	Action
1	Définissez la valeur de l'horodateur.
2	Quelques semaines plus tard, déterminez l'erreur de l'horodateur sur votre automate.
3	Calculez le facteur nécessaire pour corriger l'horodateur pour une semaine.
4	Réglez le facteur de correction de l'horodateur sur cette valeur. **Résultat :** Le facteur de correction de l'horodateur est ajouté à (ou soustrait de) la valeur de l'horodateur de votre automate chaque semaine.

Le facteur de correction de l'horodateur de Twido Extreme est une valeur de 6 bits :

Bits	Description	Valeurs possibles
0	Facteur de correction en secondes	0 à 63
1		
2		
3		
4		
5		
6	Signe du facteur de correction	0 : soustraction 1 : addition

Description des langages Twido

III

Présentation

Objet de cette partie

Cette rubrique fournit des instructions d'utilisation des langages de programmation Grafcet, schéma à contacts et liste d'instructions permettant de créer des programmes pour des automates programmables Twido.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
14	Langage schéma à contacts	385
15	Langage liste d'instructions	407
16	Grafcet	419

Langage schéma à contacts

14

Présentation

Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage schéma à contacts.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Introduction aux schémas à contacts	386
Principes de programmation en langage schéma à contacts	388
Blocs de schémas à contacts	390
Eléments graphiques du langage schéma à contacts	393
Instructions spéciales OPEN et SHORT du langage schéma à contacts	396
Conseils de programmation	398
Réversibilité schéma à contacts/liste	401
Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions	403
Documentation du programme	405

Introduction aux schémas à contacts

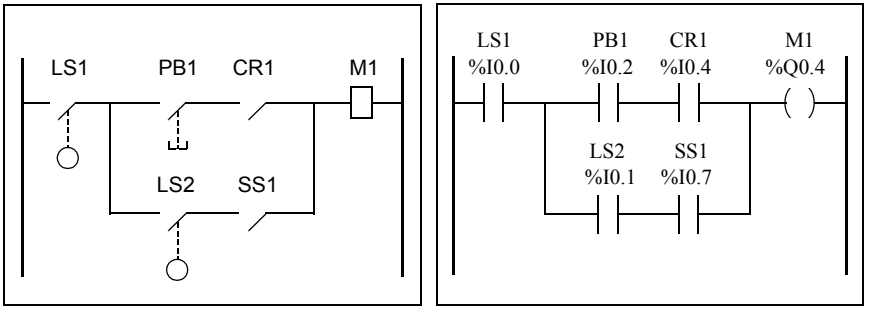
Introduction

Les schémas à contacts utilisent la même représentation graphique que celle des circuits de relais en logique programmée. à ceci près que, dans un schéma à contacts :

- Toutes les entrées sont représentées par des symboles de contacts (⊣⊢).
- Toutes les sorties sont représentées par des symboles de bobines (⊣( )⊢).
- Les opérations numériques sont comprises dans le jeu d'instructions graphiques du schéma à contacts.

Représentations de schémas à contacts correspondant aux circuits de relais

L'illustration suivante présente un schéma simplifié de câblage de relais en logique programmée, et son équivalent en langage schéma à contacts.



Circuit de relais en logique programmée

Schéma à contacts

Dans l'illustration précédente, toutes les entrées associées à un périphérique de commutation dans le circuit de relais en logique programmée sont représentées sous la forme de contacts dans le schéma à contacts. La bobine de sortie M1 du circuit logique de relais est représentée par un symbole de bobine dans le schéma à contacts. Les numéros des repères apparaissant au-dessus du symbole de chaque contact et de chaque bobine dans le schéma à contacts sont des références aux emplacements des connexions externes en entrée et en sortie vers l'automate.

Réseaux langage schéma à contacts

Un programme en langage schéma à contacts est composé de "réseaux", représentant des ensembles d'instructions graphiques et apparaissant entre deux barres verticales. Les réseaux sont exécutés de manière séquentielle par l'automate.

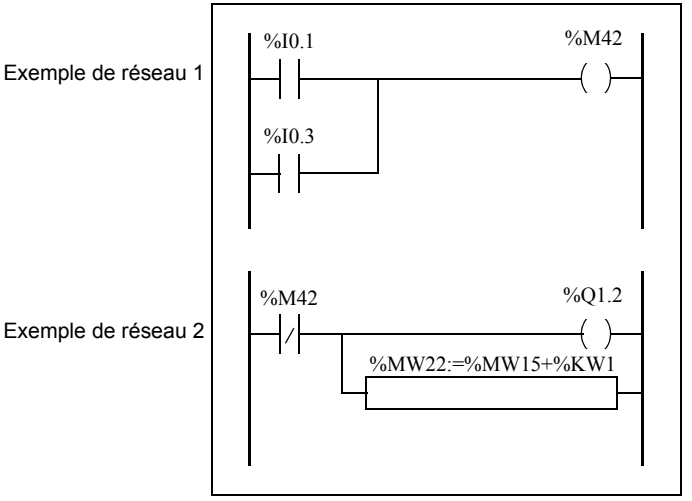
L'ensemble des instructions graphiques représente les fonctions suivantes :

- Entrées/sorties de l'automate (boutons de commande, capteurs, relais, voyants, etc.)
- Fonctions de l'automate (temporisateurs, compteurs, ...)
- Opérations mathématiques et logiques (addition, division, AND, XOR, etc.)
- Opérateurs de comparaison et autres opérations numériques (A<B, A=B, décalage, rotation, etc.)
- Variables internes de l'automate (bits, mots, etc.)

Ces instructions sont disposées graphiquement selon des connexions verticales et horizontales, débouchant éventuellement sur une ou plusieurs sorties et/ou actions. Un réseau ne peut pas contenir plus d'un groupe d'instructions liées.

Exemple de réseaux schéma à contacts

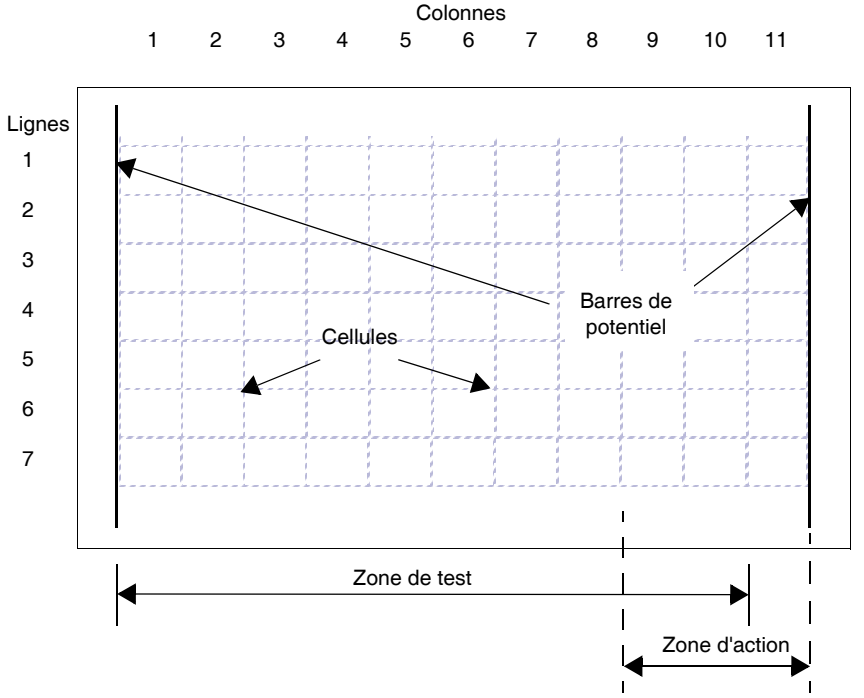
L'exemple suivant illustre un programme en langage schéma à contacts composé de deux réseaux.



Principes de programmation en langage schéma à contacts

Grille de programmation

Chaque réseau schéma à contacts se compose d'une grille comportant sept lignes et onze colonnes organisées en deux zones, comme l'indique l'illustration suivante :



Zones de la grille

La grille de programmation en langage schéma à contacts est divisée en deux zones :

- Zone de test
  Contient les conditions qui sont testées pour effectuer des actions. Se compose des colonnes 1 à 10 et contient les contacts, les blocs fonction et les blocs comparaison.
- Zone d'action
  Contient les sorties ou opérations à effectuer en fonction des résultats des tests de conditions dans la zone de test. Se compose des colonnes 8 à 11 et contient les bobines et les blocs opération.

Saisie d'instructions dans la grille

Un réseau langage schéma à contacts fournit une grille de programmation de sept lignes sur onze colonnes qui commence dans la première cellule dans le coin supérieur gauche de la grille. La programmation consiste à saisir des instructions dans les cellules de la grille. Les instructions, comparaisons et fonctions de test sont saisies dans les cellules de la zone de test et sont justifiées à gauche. La logique du test permet d'assurer la continuité dans la zone d'action où les bobines, les opérations numériques et les instructions de régulation du flux du programme sont entrées et justifiées à droite.

Le réseau est traité ou exécuté (tests effectués et sorties affectées) dans la grille de haut en bas et de gauche à droite.

Sections/Sous-programmes

Chaque section/sous-programme se compose :

- d'un en-tête de section avec un numéro de section (affecté automatiquement par le programme), une étiquette de section/sous-programme, un titre de section/sous-programme défini par l'utilisateur et quatre lignes réservées aux commentaires de l'utilisateur. (Voir Utilisation des en-têtes de section, Aide en ligne TwidoSuite).
- d'une séquence de réseaux sous l'en tête de section/sous-programme.

En-têtes réseau

En plus du réseau, un en-tête réseau apparaît directement au-dessus du réseau. Utilisez l'en-tête réseau pour documenter le rôle logique du réseau. L'en-tête réseau peut contenir les informations suivantes :

- le numéro du réseau ;
- les étiquettes (%Li) ;
- le titre du réseau ;
- des commentaires sur le réseau.

Pour obtenir davantage d'informations sur l'utilisation d'un en-tête réseau pour documenter vos programmes, reportez-vous à la rubrique *p. 405*.

Blocs de schémas à contacts

Introduction

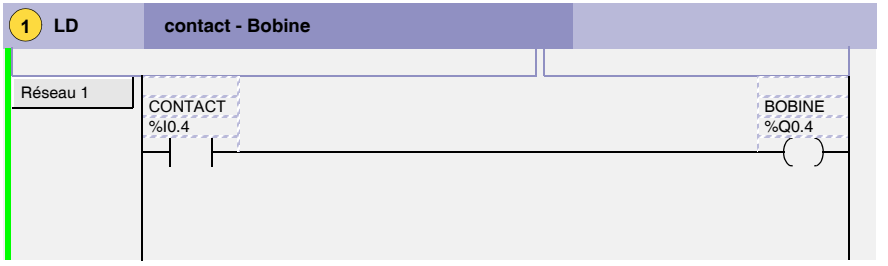
Les schémas à contacts se composent de blocs correspondant à des actions et/ou des fonctions d'un programme, telles que :

- des contacts ;
- des bobines ;
- des instructions de déroulement du programme ;
- des blocs fonction ;
- des blocs comparaison ;
- des blocs opération.

Contacts, bobines et déroulement du programme

Les contacts, bobines et les instructions de déroulement du programme (sauts et appels) n'occupent qu'une seule cellule dans la grille de programmation du schéma à contacts. Les blocs fonction, les blocs comparaison et les blocs opération peuvent en revanche occuper plusieurs cellules.

Les exemples suivants illustrent un contact et une bobine.

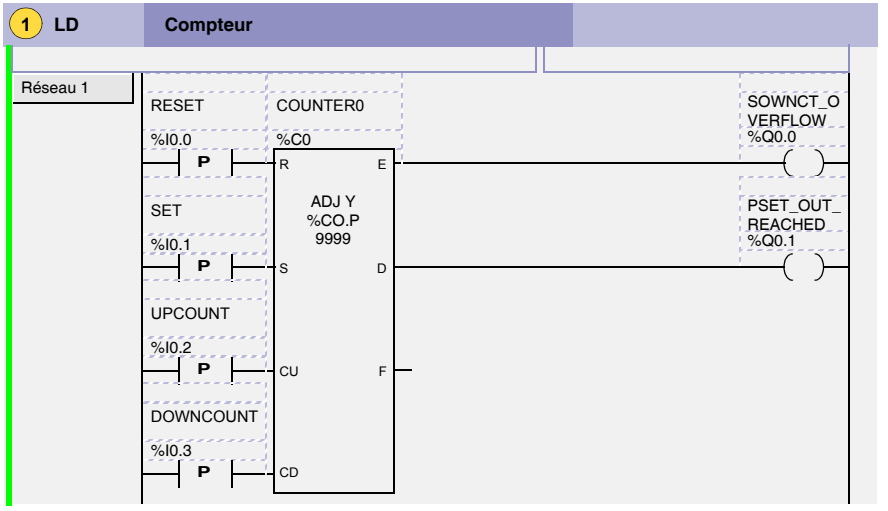


Blocs fonction

Les blocs fonction sont placés dans la zone de test de la grille de programmation. Le bloc doit apparaître dans la première ligne, aucune instruction de schéma à contacts ni aucune ligne de continuité ne pouvant apparaître au-dessus ou au-dessous du bloc fonction. Les instructions de test des schémas à contacts mènent à l'entrée du bloc fonction, tandis que les instructions de test et/ou d'action mènent à la sortie du bloc fonction.

Les blocs fonction sont orientés de manière verticale et occupent deux colonnes sur quatre lignes dans la grille de programmation.

L'exemple suivant illustre un bloc fonction temporisateur.

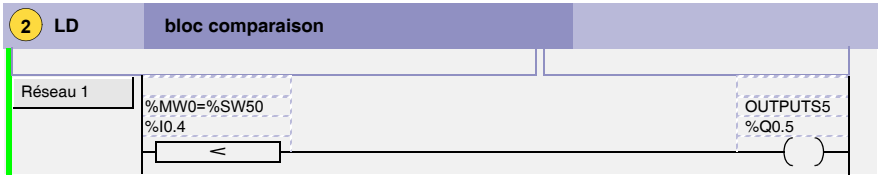


Blocs comparaison

Les blocs comparaison sont placés dans la zone de test de la grille de programmation. Le bloc peut apparaître dans n'importe quelle ligne ou colonne de la zone de test. L'intégralité de l'instruction doit résider dans cette zone.

Les blocs comparaison sont orientés de manière horizontale et occupent deux colonnes sur une ligne dans la grille de programmation.

L'exemple suivant présente un bloc comparaison.



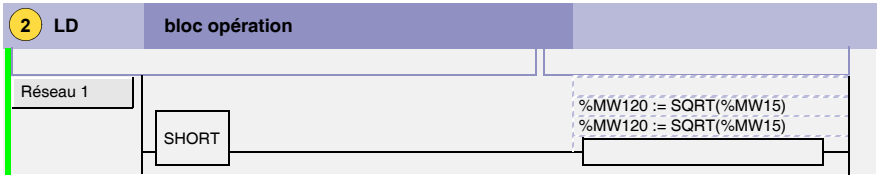
Pour modifier le bloc comparaison, cliquez sur le champ situé au-dessus de la zone de comparaison et saisissez votre expression. Des symboles et des adresses peuvent être utilisés en tant qu'opérandes. Le champ de commentaires est désactivé.

Blocs opération

Les blocs opération sont placés dans la zone d'action de la grille de programmation. Le bloc peut apparaître dans n'importe quelle ligne de la zone d'action. L'instruction est justifiée à droite ; elle apparaît sur la droite et se termine dans la dernière colonne.

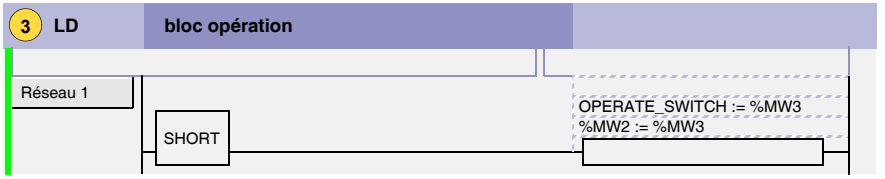
Les blocs opération sont orientés de manière horizontale et occupent quatre colonnes sur une ligne dans la grille de programmation.

L'exemple suivant illustre un bloc opération.



Pour modifier le bloc opération, cliquez sur le champ situé au-dessus de la zone d'opération et entrez votre expression. Des symboles et des adresses peuvent être utilisés en tant qu'opérandes. L'expression s'affiche deux fois, comme illustré dans l'exemple ci-avant. Le champ de commentaires (zone du haut) est désactivé.

Si des symboles ont été précédemment définis, l'expression s'affiche avec les adresses (zone du bas) et les symboles (zone du haut), comme illustré dans l'exemple de bloc opération ci-après. Dans cet exemple, l'adresse %MW2 a été définie avec le symbole OPERATE_SWITCH.



Eléments graphiques du langage schéma à contacts

Introduction

Les instructions des schémas à contacts sont constituées d'éléments graphiques.

Contacts

Les éléments graphiques des contacts sont programmés dans la zone de test et occupent une cellule (une ligne sur une colonne).

Nom	Elément graphique	Instruction	Fonction
Contact à ouverture		LD	Contact passant lorsque l'objet bit de contrôle se trouve à l'état 1.
Contact à fermeture		LDN	Contact passant lorsque l'objet bit de contrôle se trouve à l'état 0.
Contact de détection d'un front montant		LDR	Front montant : détecte le passage de 0 à 1 de l'objet bit de contrôle.
Contact de détection d'un front descendant		LDF	Front descendant : détecte le passage de 1 à 0 de l'objet bit de contrôle.

Eléments de liaison

Les éléments de liaison graphique permettent d'insérer/de supprimer des boucles de schéma à contacts :

Nom	Elément graphique	Fonction
Insertion d'un lien		Insertion d'une boucle de schéma à contacts vide
Suppression d'un lien		Suppression d'une boucle de schéma à contacts vide **Remarque :** **1.** Si la boucle de schéma à contacts contient des éléments, vous devez les supprimer avant de pouvoir supprimer la boucle elle-même. **2.** Le raccourci clavier utilisé pour supprimer un élément est SUPPR.

Bobines

Les éléments graphiques des bobines sont programmés dans la zone d'action et occupent une cellule (une ligne sur une colonne).

Nom	Elément graphique	Instruction	Fonction
Bobine directe		ST	L'objet bit associé prend la valeur du résultat de la zone de test.
Bobine inverse		STN	L'objet bit associé prend la valeur du résultat inverse de la zone de test.
Bobine d'enclenchement		S	L'objet bit associé est réglé sur 1 lorsque le résultat de la zone de test est 1.
Bobine de déclenchement		R	L'objet bit associé est réglé sur 0 lorsque le résultat de la zone de test est 1.
Appel de saut ou de sous-programme	->>%Li ->>%SRi	JMP SR	Se connecte à une instruction portant une étiquette, en amont ou en aval.
Bobine dièse			Langage Grafcet. Utilisée lorsque la programmation des conditions de transition associées aux transitions provoque une permutation sur l'étape suivante.
Retour d'un sous-programme	<RET>	RET	Placé à la fin des sous-programmes pour retourner au programme principal.
Arrêt du programme	<END>	END	Définit la fin du programme.

Blocs fonction

Les éléments graphiques des blocs fonction sont programmés dans la zone de test et occupent quatre lignes sur deux colonnes (excepté les compteurs rapides (VFC), qui requièrent cinq lignes sur deux colonnes).

Nom	Elément graphique	Fonction
Temporisateurs, compteurs, registres, etc.		Chaque bloc fonction utilise les entrées et les sorties permettant la liaison aux autres éléments graphiques. Remarque : Les sorties des blocs fonction ne peuvent pas être connectées les unes aux autres (liaisons verticales).

Blocs opération et comparaison

Les blocs comparaison sont programmés dans la zone de test et les blocs opération, dans la zone d'action.

Nom	Elément graphique	Fonction
Bloc comparaison		Compare deux opérandes. La sortie prend la valeur 1 lorsque le résultat est vérifié. Taille : Une ligne sur deux colonnes
bloc opération		Effectue des opérations arithmétiques et logiques. Taille : Une ligne sur quatre colonnes

Instructions spéciales OPEN et SHORT du langage schéma à contacts

Introduction

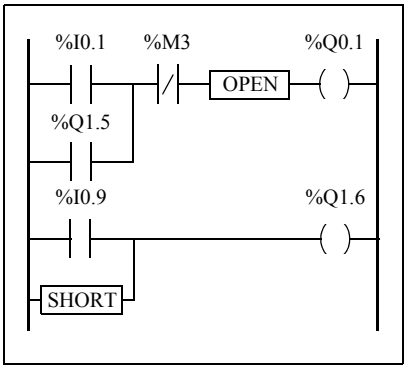
Les instructions OPEN et SHORT proposent une méthode de mise au point et de dépannage facile des programmes en langage schéma à contacts. Ces instructions spéciales modifient la logique d'un réseau soit en raccourcissant soit en ouvrant la continuité du réseau comme l'explique le tableau suivant :

Instruction	Description	Instruction en langage liste d'instructions
OPEN	Au début du réseau.	`LD 0`
	Dans un réseau. Crée un arrêt dans la continuité d'un réseau schéma à contacts, et ce, quels que soient les résultats de la dernière opération logique.	`AND 0`
SHORT	Au début du réseau.	`LD 1`
	Dans un réseau. Permet à la continuité de traverser le réseau schéma à contacts, et ce, quels que soient les résultats de la dernière opération logique.	`OR 1`

En langage liste d'instructions, les instructions LD, OR et AND sont utilisées pour créer les instructions OPEN et SHORT à l'aide des valeurs immédiates respectives de 0 et 1.

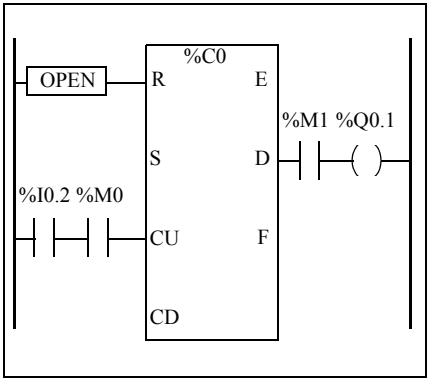
Exemples

Les exemples suivants illustrent l'utilisation des instructions SHORT et OPEN.




```
LD      %I0.1
OR      %Q1.5
ANDN    %M3
AND     0
ST      %Q0.1
LD      %I0.9
OR      1
ST      %Q1.6
```

Lorsqu'aucune entrée de réinitialisation n'est requise, utilisez l'élément OPEN pour interrompre définitivement la liaison vers le segment d'entrée. Cette opération est nécessaire car vous n'avez pas le droit de supprimer le segment assurant la liaison vers l'entrée R du bloc fonction.



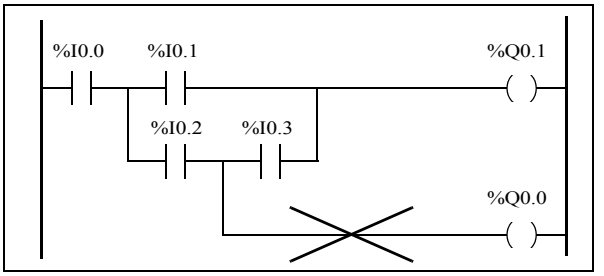

```
BLK       %C0
LD        0
R
LD        %I0.2
AND       %M0
CU
OUT_BLK
LD        D
AND       %M1
ST        %Q0.1
END_BLK
```

Conseils de programmation

Gestion des sauts de programme	Utilisez les sauts de programme avec la plus grande précaution, car ils peuvent être à l'origine de boucles qui ralentiront considérablement les opérations de scrutation. Evitez d'insérer des sauts pointant vers des instructions situées en amont. (Une instruction en amont apparaît avant un saut dans un programme. A l'inverse, une instruction en aval apparaît après un saut dans un programme).
Programmation des sorties	Les bits de sortie, tout comme les bits internes, ne doivent être modifiés qu'une seule fois dans le programme. Pour les bits de sortie, seule la dernière valeur scrutée est prise en compte lors de la mise à jour des sorties.
Utilisation de capteurs d'arrêt d'urgence à liaison directe	Les capteurs utilisés en cas d'arrêt d'urgence ne doivent pas être gérés par l'automate. Ces capteurs doivent être raccordés directement aux sorties correspondantes.
Gestion des reprises de l'alimentation	Conditionner une reprise secteur à une opération manuelle. Un redémarrage automatique peut entraîner un fonctionnement non désiré de l'installation (utilisez les bits système %S0, %S1 et %S9).
Gestion de l'heure et des blocs horodateur	Il est nécessaire de vérifier l'état du bit système %S51, qui signale d'éventuels défaut de l'horodateur.
Vérification de la syntaxe et recherche d'erreurs	Lors de la saisie d'un programme, TwidoSuite vérifie la syntaxe de ses instructions et opérandes, ainsi que leur association.

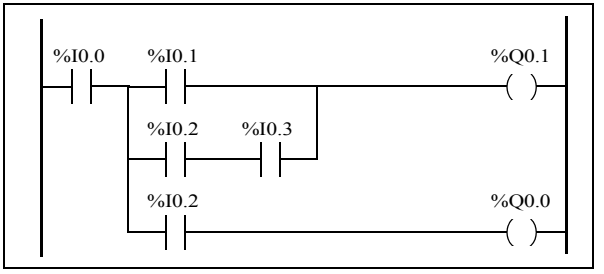
Remarques complémentaires sur l'utilisation des parenthèses

Les opérations d'affectation ne doivent pas être placés entre parenthèses :



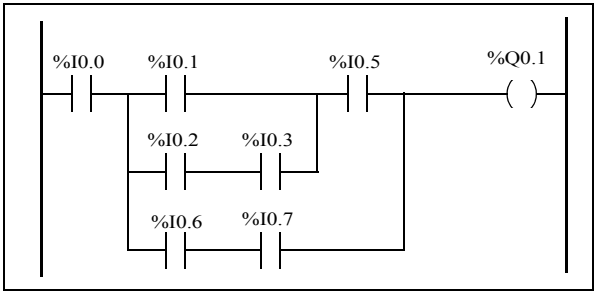

```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
ST      %Q0.0
AND     %I0.3
)
ST      %Q0.1
```

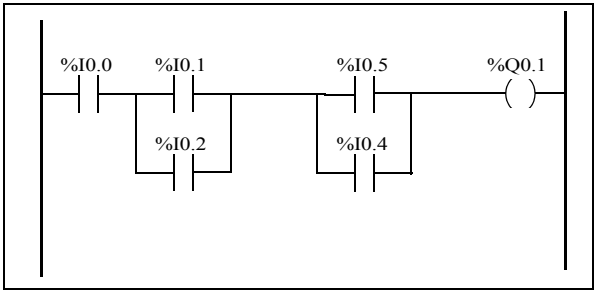
Afin d'effectuer la fonction correspondante, les équations suivantes doivent être programmées :



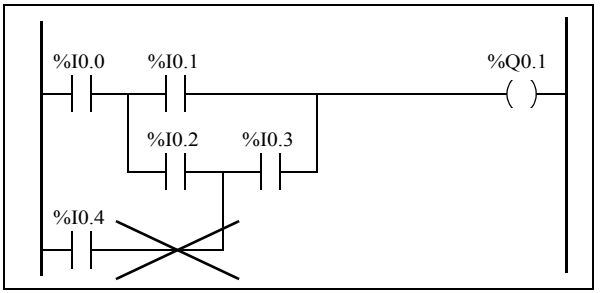

```
LD      %I0.0
MPS
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
ST      %Q0.1
MPP
AND     %I0.2
ST      %Q0.0
```

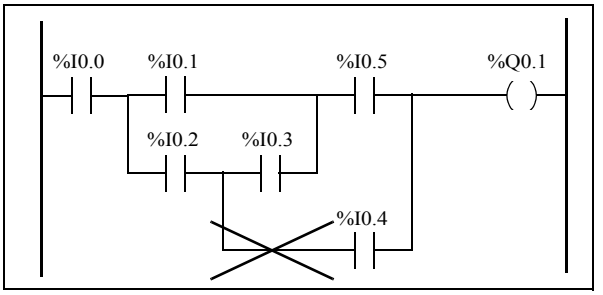
Si plusieurs mises en parallèle de contact sont effectuées, elles devront être imbriquées les unes dans les autres ou complètement dissociées :



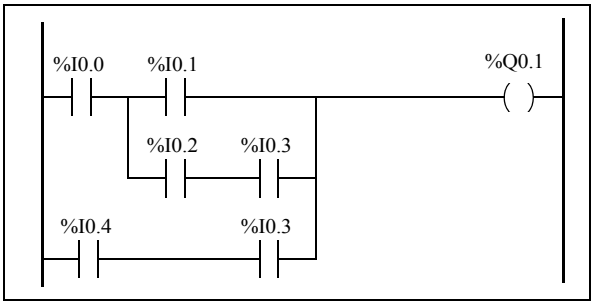


Les schémas suivants ne peuvent pas être programmés :

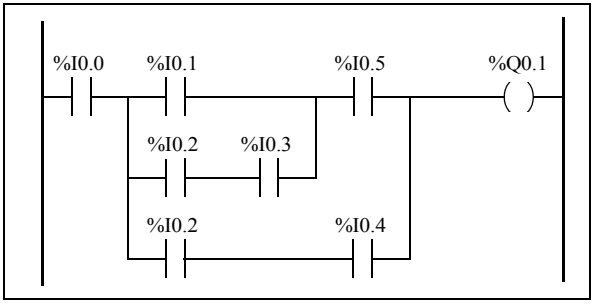




Afin d'exécuter les schémas équivalents, modifiez-les comme illustré ci-dessous :




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
OR(     %I0.4
AND     %I0.3
)
ST      %Q0.1
```




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
AND     %I0.5
OR(     %I0.2
AND     %I0.4
)
)
ST      %Q0.1
```

Réversibilité schéma à contacts/liste d'instructions

Présentation

La fonction de réversibilité du logiciel de programmation TwidoSuite permet de convertir des parties de programmes par schémas à contacts en parties de programmes par listes d'instructions, et vice versa.

Vous pouvez régler l'affichage par défaut des programmes soit au format liste d'instructions, soit au format schéma à contacts, dans les préférences (Préférences, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) utilisateur. Vous pouvez basculer des sections de programmes entre les formats listes d'instructions et schémas à contacts.

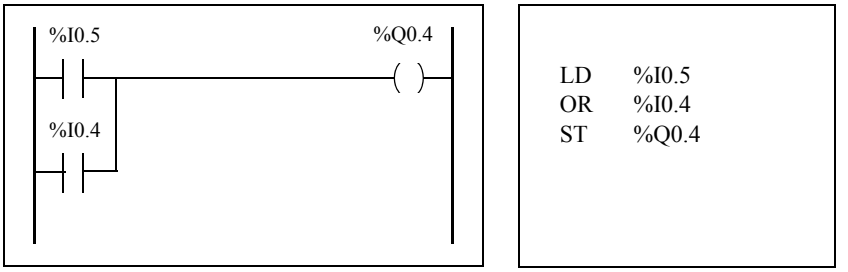
Note : Lors du basculement entre les affichages en listes d'instructions et en schémas à contacts, seuls les réseaux de la section sélectionnée sont inversés. Ceci vous permet d'afficher un programme avec certaines sections au format schéma à contacts et d'autres au format liste d'instructions dans la même fenêtre.

Qu'est-ce que la "réversibilité" ?

Pour bien comprendre à quoi correspond la fonction de réversibilité du programme, il convient d'examiner avec attention les relations existant entre le réseau d'un schéma à contacts et la séquence de la liste d'instructions correspondante :

- **Réseau de schéma à contacts** : ensemble d'instructions par schémas à contacts formant une expression logique.
- **Séquence de liste** : ensemble d'instructions d'un programme par listes, correspondant aux instructions par schémas à contacts et relatif à la même expression logique.

L'illustration suivante présente un réseau de schéma à contacts courant, ainsi que la logique du programme équivalente, exprimée sous la forme d'une liste d'instructions.



Un programme d'application est stocké en interne sous la forme d'une liste d'instructions et ce, que le programme ait été rédigé en langage à schéma à contacts ou en langage listes d'instructions. TwidoSuite utilise les similarités de structure de programme existant entre les deux langages, ainsi que l'image liste interne du programme pour l'afficher soit sous la forme d'une liste d'instructions (forme élémentaire), soit de manière graphique, sous la forme d'un schéma à contacts, en fonction des préférences sélectionnées par l'utilisateur.

Garantie de réversibilité

Tout programme créé sous forme de schéma à contacts peut être converti en une liste d'instructions. En revanche, certaines logiques du langage par listes ne peuvent pas être converties en langage à schéma à contacts. Pour garantir une réversibilité totale entre le langage listes d'instructions et le langage à schéma à contacts, il est important d'observer les directives présentées à la section *p. 403*.

Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions

Instructions requises pour la réversibilité

La structure d'un bloc fonction réversible dans le langage liste d'instructions requiert l'utilisation des instructions suivantes :

- **BLK** marque le début du bloc et définit le début du réseau, ainsi que celui de la portion d'entrée dans le bloc.
- **OUT_BLK** marque le début de la portion de sortie du bloc.
- **END_BLK** marque la fin du bloc et du réseau.

Il n'est pas nécessaire d'utiliser des instructions de blocs fonction réversibles pour un programme liste d'instructions qui fonctionne correctement. Certaines instructions permettent une programmation liste d'instructions non réversible. Pour obtenir des informations complètes sur la programmation liste d'instructions non réversible de blocs fonction, reportez-vous à la rubrique *p. 533*.

Instructions sans équivalences à éviter

Evitez d'utiliser certaines instructions en langage liste ou certaines associations d'instructions et d'opérandes, pour lesquelles les schémas à contacts ne possèdent pas d'équivalents. Par exemple, l'instruction N (permettant d'inverser la valeur de l'accumulateur booléen) n'a pas d'équivalent dans le langage schémas à contacts.

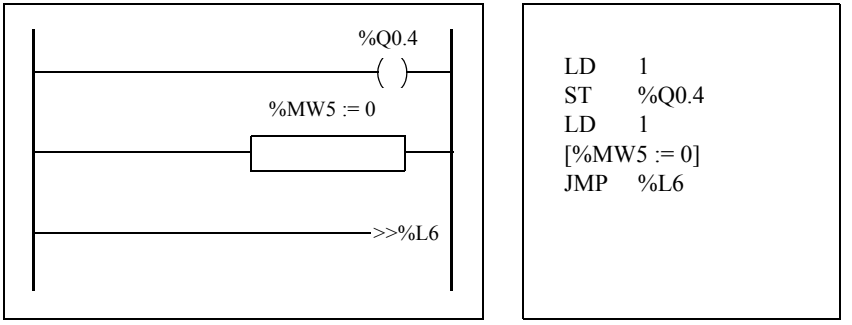
Le tableau suivant répertorie toutes les instructions de programmation liste d'instructions qui ne s'inversent pas dans le langage schéma à contacts :

Instruction par liste	Opérande	Description
JMPCN	%Li	Not saut conditionnel
N	aucun	Négation (Not)
ENDCN	aucun	Not fin conditionnelle

Réseaux inconditionnels

La programmation des réseaux inconditionnels requiert également l'application des recommandations de programmation liste d'instructions pour que la réversibilité liste d'instructions/schéma à contacts puisse s'opérer. Les réseaux inconditionnels ne sont pas soumis à des tests ou à des conditions. Les sorties ou les instructions d'action sont toujours exécutées.

Le diagramme suivant présente des exemples de réseaux inconditionnels, ainsi que la séquence en langage liste d'instructions équivalente.



Vous noterez que chacune des séquences en langage liste d'instructions inconditionnelles ci-dessus commence par une instruction de chargement suivie d'un 1, excepté pour l'instruction JMP. Cette combinaison règle la valeur de l'accumulateur booléen sur 1, et règle par conséquent la bobine (instruction de stockage) sur 1 et %MW5 sur 0 lors de chaque scrutation du programme. L'exception est l'instruction de saut inconditionnel en langage liste d'instructions (JMP %L6) qui est exécutée quelle que soit la valeur de l'accumulateur et ne nécessite pas le réglage de l'accumulateur sur 1.

Réseau schéma à contacts / liste d'instructions

Si un programme liste d'instructions qui n'est pas totalement réversible est inversé, les parties réversibles sont affichées dans la visualisation par schémas à contacts et celles qui sont irréversibles sont affichées sur les réseaux schéma à contacts en liste d'instructions.

Un réseau schéma à contacts en liste d'instructions fonctionne exactement comme un petit éditeur liste d'instructions. Il permet en effet à l'utilisateur de visualiser et de modifier les parties irréversibles d'un programme schéma à contacts.

Documentation du programme

Documentation de votre programme

Vous pouvez documenter votre programme en y ajoutant des commentaires à l'aide des éditeurs liste d'instructions et schéma à contacts :

- Dans l'éditeur de listes, des commentaires de lignes vous permettent de documenter votre programme. Ces commentaires peuvent figurer sur la même ligne que les instructions de programmation, ou sur des lignes individuelles distinctes.
- Dans l'éditeur de schémas à contacts, des en-têtes réseau vous permettent de documenter votre programme. Ces en-têtes se situent juste au-dessus du réseau.

Le logiciel de programmation TwidoSuite utilise ces commentaires à des fins de réversibilité. Lors de la conversion d'un programme par listes en programme par schémas à contacts, TwidoSuite utilise certains des commentaires liste pour créer un en-tête réseau. Pour ce faire, les commentaires insérés entre les séquences de liste sont utilisés comme en-têtes réseau.

Exemple de commentaires de ligne de liste

L'exemple suivant illustre un programme par listes comportant des commentaires de lignes.

```
---- ( * TITRE DE L'EN-TETE DU RESEAU 0 * )
---- ( * PREMIER COMMENTAIRE DE L'EN-TETE DU RESEAU 0 * )
---- ( * DEUXIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 0 * )
      0   LD  % I0. 0  ( * LIGNE DE COMMENTAIRE *)
      1   OR  %I0. 1  ( * LIGNE DE COMMENTAIRE IGNOREE LORS DE LA CONVERSION EN LANGAGE
SCHEMA À CONTACTS * )
      2   ANDM  %M10
      3   ST        M101
---- ( * EN-TETE DU RESEAU 1 * )
---- ( * CE RESEAU CONTIENT UNE ETIQUETTE * )
---- ( * DEUXIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
---- ( * TROISIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
---- ( * QUATRIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
      4   % L5:
      5   LD  %M101
      6   [ %MW20 := %KW2 * 16]
---- ( * CE RESEAU NE CONTIENT QUE LE TITRE D'UN EN-TETE * )
      7   LD  %Q0. 5
      8   OR  %I0.  3
      9   ORR  I0.  13
     10   ST  %Q0.5
```

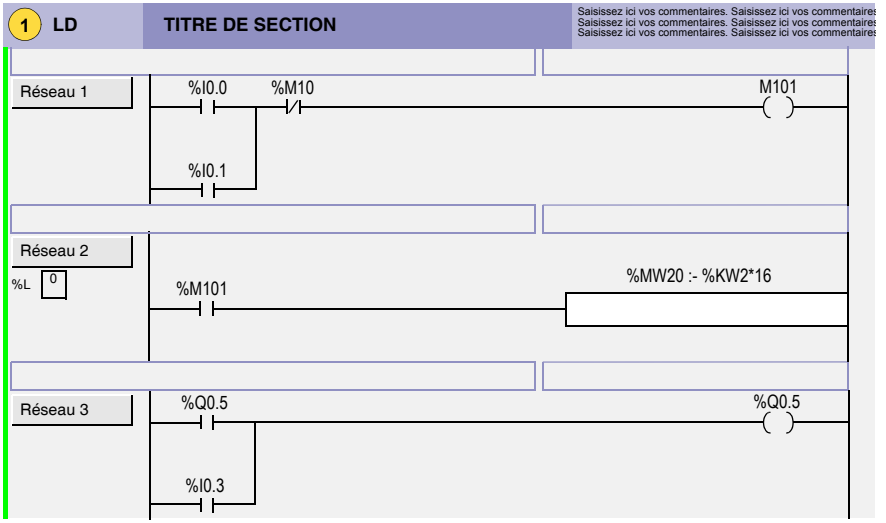
Conversion de commentaires de liste en en-tête réseau de schéma à contacts

Lorsque qu'un programme par listes est converti en programme par schémas à contacts, les commentaires de ligne de liste sont affichés dans l'éditeur de schémas à contacts selon les règles suivantes :

- Le premier commentaire figurant sur une ligne individuelle est utilisé comme en-tête réseau.
- Les commentaires suivants sont utilisés pour former le corps du réseau.
- Lorsque les lignes du corps de l'en-tête sont toutes remplies, les commentaires de ligne compris entre les séquences de liste sont ignorés, tout comme les autres commentaires situés dans des lignes de liste et qui contiennent également des instructions.

Exemple de commentaires d'en-têtes réseau

L'exemple suivant illustre un programme par schémas à contacts comportant des commentaires d'en-têtes réseau.



Conversion de commentaires de schémas à contacts en commentaires de listes

Lorsqu'un schéma à contacts est converti en une liste d'instructions, les commentaires d'en-têtes réseau sont affichés dans l'éditeur de listes selon les règles suivantes :

- Tous les commentaires d'en-tête réseau sont insérés entre les séquences de liste associées.
- Toutes les étiquettes (%Li:) ou les déclarations de sous-programme (SRi:) sont placées sur la ligne suivant l'en-tête et précédant immédiatement la séquence de liste.
- Si le programme avait déjà été converti du format liste au format schéma à contacts, tous les commentaires précédemment ignorés seront de nouveau affichés dans l'éditeur de listes.

Langage liste d'instructions

15

Présentation

Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage liste d'instructions.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble des programmes en langage liste d'instructions	408
Fonctionnement des listes d'instructions	410
Instructions en langage liste d'instructions	411
Utilisation de parenthèses	414
Instructions de pile (MPS, MRD, MPP)	417

Vue d'ensemble des programmes en langage liste d'instructions

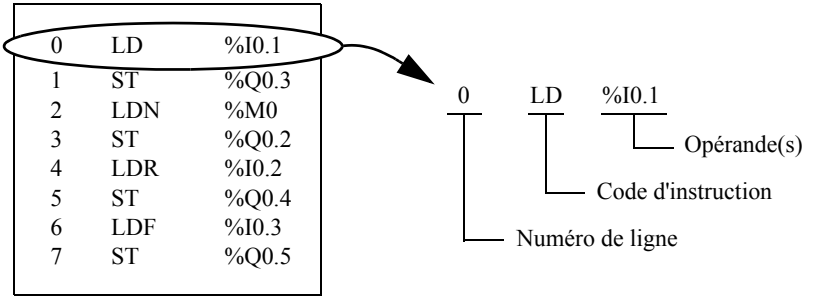
Introduction

Un programme écrit en langage liste d'instructions est constitué d'une série d'instructions exécutées en séquence par l'automate. Chaque instruction est représentée par une seule ligne de code et se compose de trois éléments :

- Numéro de ligne
- Code d'instruction
- Opérande(s)

Exemple de programme liste d'instructions

L'illustration suivante est un exemple de programme liste d'instructions.



Numéro de ligne

Les numéros de ligne sont générés automatiquement lorsque vous saisissez une instruction. Les lignes vides et les lignes de commentaires n'ont pas de numéro de ligne.

Code d'instruction

Le code d'instruction est un symbole désignant un opérateur qui identifie l'opération à effectuer à l'aide des opérandes. Les opérateurs types spécifient les opérations booléennes et numériques.

Par exemple, dans l'échantillon de programme présenté ci-dessus, LD est l'abréviation de LOAD en code d'instruction. L'instruction LOAD place (charge) la valeur de l'opérande %I0.1 dans un registre interne nommé accumulateur.

Il existe deux types d'instructions de base :

- Instructions de test
  Il s'agit de tests des conditions ou résultats d'équation nécessaires à l'accomplissement d'une action. Par exemple, LOAD (LD) et AND.
- Instructions d'action
  Elles permettent d'effectuer les actions autorisées lorsque les tests de configuration sont remplies. Par exemple, des instructions d'affectation telles que STORE (ST) et RESET (R).

Opérande

Un opérande est un nombre, un repère ou un symbole représentant une valeur qu'un programme peut manipuler au sein d'une instruction. Par exemple, dans l'échantillon de programme présenté ci-dessus, l'opérande %I0.1 est un repère auquel on a affecté la valeur d'une entrée de l'automate. Une instruction peut avoir entre zéro et trois opérandes selon le type de code d'instruction.

Les opérandes peuvent représenter les éléments suivants :

- les entrées/sorties de l'automate, telles que les capteurs, boutons poussoirs et relais ;
- les fonctions système prédéfinies, telles que les temporisateurs et les compteurs ;
- les opérations arithmétiques, logiques, de comparaisons et numériques ;
- les variables internes de l'automate, telles que les bits et les mots.

Fonctionnement des listes d'instructions

Introduction

Les listes d'instructions ne possèdent qu'un seul opérande explicite, l'autre étant implicite. L'opérande implicite correspond à la valeur de l'accumulateur booléen. Par exemple, dans l'instruction LD %I0.1, %I0.1 est l'opérande explicite. Un opérande implicite est stocké dans l'accumulateur et se voit écrasé par la valeur de %I0.1.

Fonctionnement

Une instruction en langage liste d'instructions exécute une opération spécifiée sur le contenu de l'accumulateur et sur l'opérande explicite, puis remplace le contenu de l'accumulateur par le résultat obtenu. Par exemple, l'opération AND %I1.2 effectue un AND logique entre le contenu de l'accumulateur et celui de l'entrée 1.2 et remplace le contenu de l'accumulateur par ce résultat.

L'ensemble des instructions booléennes, à l'exception des instructions de chargement, de stockage et les instructions NOT, fonctionnent avec deux opérandes. La valeur des deux opérandes peut être True ou False et l'exécution des instructions par le programme génère une valeur unique : soit True, soit False. Les instructions de chargement placent la valeur de l'opérande dans l'accumulateur, tandis que les instructions de stockage transfèrent la valeur de l'accumulateur vers l'opérande. L'instruction NOT ne comporte aucun opérande explicite et a seulement pour effet d'inverser l'état de l'accumulateur.

Instructions en langage liste d'instructions prises en charge

Le tableau suivant représente quelques instructions en langage liste :

Type d'instruction	Exemple	Fonction
Instruction sur bit	LD %M10	Lit le bit interne %M10
Instruction sur bloc	IN %TM0	Démarre le temporisateur %TM0
Instruction sur mot	[%MW10 := %MW50+100]	Opération d'addition
Instruction sur programme	SR5	Appelle le sous-programme n°5
Instruction Grafcet	-*-8	Etape n°8

Instructions en langage liste d'instructions

Introduction

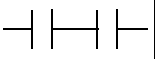
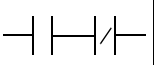
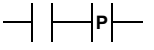
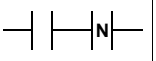
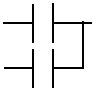
Le langage liste d'instructions comprend les types d'instructions suivants :

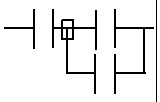
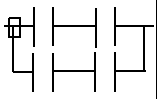
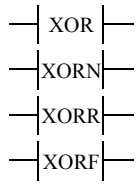
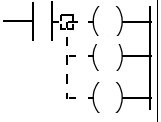
- Instructions sur test
- Instructions sur action
- Instructions sur bloc fonction

Ce sous-chapitre identifie et décrit les instructions Twido de programmation en langage liste d'instructions.

Instructions sur test

Le tableau suivant décrit les instructions sur test du langage liste d'instructions.

Nom	Elément graphique correspondant	Fonction
LD		Le résultat booléen correspond à l'état de l'opérande.
LDN		Le résultat booléen correspond à l'état inversé de l'opérande.
LDR		Le résultat booléen prend la valeur 1 lorsque le passage de l'opérande (front montant) de 0 à 1 est détecté.
LDF		Le résultat booléen devient 1 lorsque le passage de l'opérande (front descendant) de 1 à 0 est détecté.
AND		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et l'état de l'opérande.
ANDN		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et l'état inversé de l'opérande.
ANDR		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et la détection du front montant de l'opérande (1 = front montant).
ANDF		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et la détection du front descendant de l'opérande (1 = front descendant).
OR		Le résultat booléen est égal à la logique OR entre le résultat booléen de l'instruction précédente et l'état de l'opérande.

Nom	Elément graphique correspondant	Fonction
AND(		Logique AND (8 niveaux de parenthèses)
OR(		Logique OR (8 niveaux de parenthèses)
XOR, XORN, XORR, XORF		OR exclusif
MPS MRD MPP		Commutation vers les bobines
N	-	Négation (NOT)

Instructions sur action

Le tableau suivant décrit les instructions sur action du langage liste d'instructions.

Nom	Elément graphique correspondant	Fonction
ST	—( )—	L'opérande associé prend la valeur du résultat de la zone de test.
STN	—(/)—	L'opérande associé prend la valeur inversée du résultat de la zone de test.
S	—(S)—	L'opérande associé est réglé sur 1 lorsque le résultat de la zone de test est 1.
R	—(R)—	L'opérande associé est réglé sur 0 lorsque le résultat de la zone de test est 1.
JMP	->>%Li	Se connecte inconditionnellement à une séquence portant une étiquette, en amont ou en aval.
SRn	->>%SRi	Connexion au début d'un sous-programme.
RET	<RET>	Retour d'un sous-programme.
END	<END>	Fin de programme.
ENDC	<ENDC>	Fin du programme conditionné avec un résultat booléen de 1.
ENDCN	<ENDCN>	Fin du programme conditionné avec un résultat booléen de 0.

Instructions de blocs fonction

Le tableau suivant décrit les instructions sur bloc fonction du langage liste d'instructions.

Nom	Elément graphique correspondant	Fonction
Temporisateurs, compteurs, registres, etc.		Il existe des instructions de régulation de bloc pour chaque bloc fonction. Une forme structurée est utilisée pour raccorder directement les entrées et les sorties du bloc. **Remarque :** Les sorties des blocs fonction ne peuvent pas être connectées les unes aux autres (liaisons verticales).

Utilisation de parenthèses

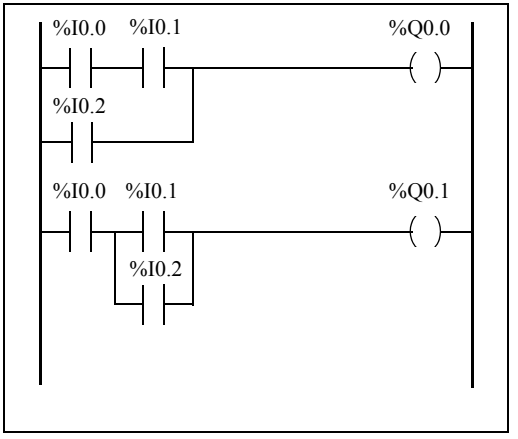
Introduction

Dans les instructions logiques AND et OR, les parenthèses permettent de spécifier des divergences dans des schémas à contacts. Les parenthèses sont associées à des instructions, de la manière suivante :

- L'ouverture des parenthèses est associée à l'instruction AND ou OR.
- La fermeture des parenthèses correspond à une instruction requise pour chaque parenthèse ouverte.

Exemple d'utilisation d'une instruction AND

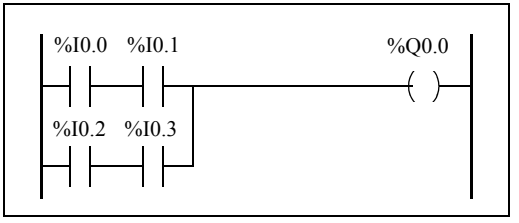
Les schémas suivants illustrent l'utilisation des parenthèses dans une instruction AND : AND(...).




```
LD      %I0.0
AND     %I0.1
OR      %I0.2
ST      %Q0.0

LD      %I0.0
AND(    %I0.1
OR      %I0.2
)
ST      %Q0.1
```

Exemple d'utilisation d'une instruction OR

Les schémas suivants illustrent l'utilisation des parenthèses dans une instruction OR : OR(...).




```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
AND     %I0.3
)
ST      %Q0.0
```

Modificateurs

Le tableau suivant répertorie les modificateurs pouvant être associés à des parenthèses.

Modificateur	Fonction	Exemple
N	Négation	AND(N ou OR(N
F	Front descendant	AND(F ou OU(F
R	Front montant	AND(R ou OU(R
[	Comparaison	Reportez-vous à la rubrique *p. 481*

Imbrication de parenthèses

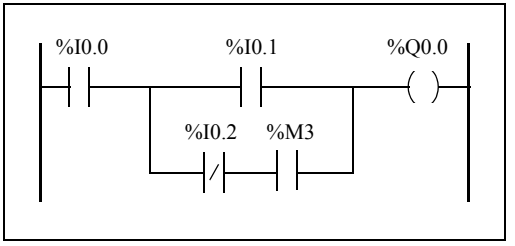
Il est possible d'imbriquer un maximum de huit niveaux de parenthèses.

Veuillez appliquer les règles suivantes lors de l'imbrication de parenthèses :

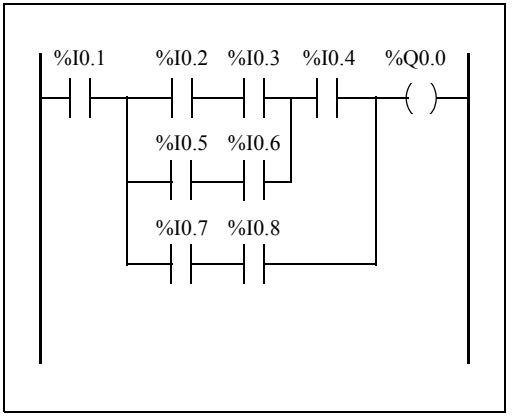
- Chaque parenthèse ouverte doit être obligatoirement refermée.
- Les étiquettes (%Li:), les sous-programmes (SRi:), les instructions de saut (JMP) et les instructions de bloc fonction ne doivent pas être placés dans des expressions comprises entre parenthèses.
- Les instructions de stockage ST, STN, S et R ne doivent pas être programmées entre parenthèses.
- Les instructions de pile MPS, MRD et MPP ne peuvent pas être utilisées entre parenthèses.

Exemples d'imbrication de parenthèses

Les schémas suivants illustrent l'imbrication de parenthèses.




```
LD      %I0.0
AND(    %I0.1
OR(N    %I0.2
AND     %M3
)
)
ST      %Q0.0
```




```
LD      %I0.1
AND(    %I0.2
AND     %I0.3
OR(     %I0.5
AND     %I0.6
)
AND     %I0.4
OR(     %I0.7
AND     %I0.8
)
)
ST      %Q0.0
```

Instructions de pile (MPS, MRD, MPP)

Introduction

Les instructions de pile permettent de traiter le routage vers des bobines .Les instructions MPS, MRD et MPP utilisent une zone de stockage temporaire appelée " pile ". Cette pile peut stocker un maximum de huit expressions booléennes.

Note : Ces instructions ne peuvent pas être utilisées dans une expression comprise entre parenthèses.

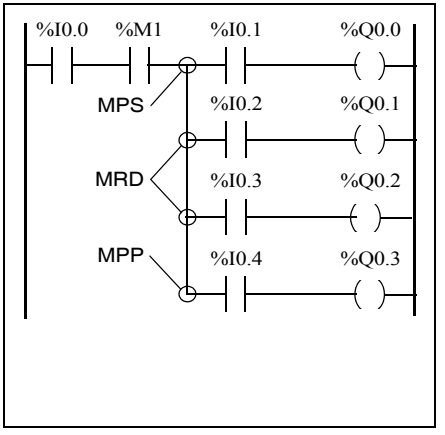
Fonctionnement des instructions de pile

Le tableau suivant décrit le fonctionnement des trois instructions de pile.

Instruction	Description	Fonction
MPS	Abréviation de Memory Push onto Stack (Mise en mémoire sur la pile)	Stocke le résultat de la dernière instruction logique (contenu de l'accumulateur) en haut de la pile. Ceci a pour effet de décaler les autres valeurs de la pile vers le bas.
MRD	Abréviation de Memory Read from stack (Lecture mémoire depuis la pile)	Lit la valeur stockée en haut de la pile et la transmet à l'accumulateur.
MPP	Abréviation de Memory Pop from Stack (Extraction mémoire depuis la pile)	Lit la valeur située dans le haut de la pile, la transmet à l'accumulateur et déplace les autres valeurs de la pile vers le haut.

Exemples d'instructions de pile

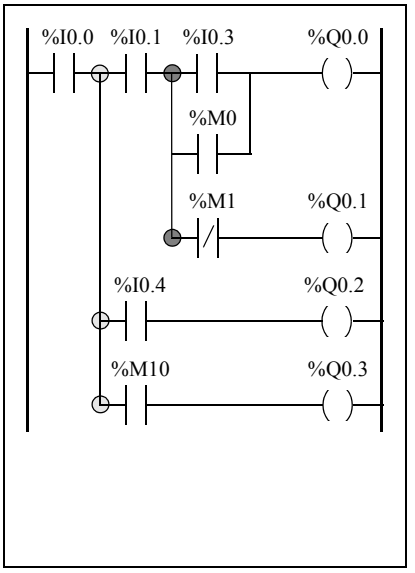
Les schémas suivants illustrent l'utilisation d'instructions de pile.

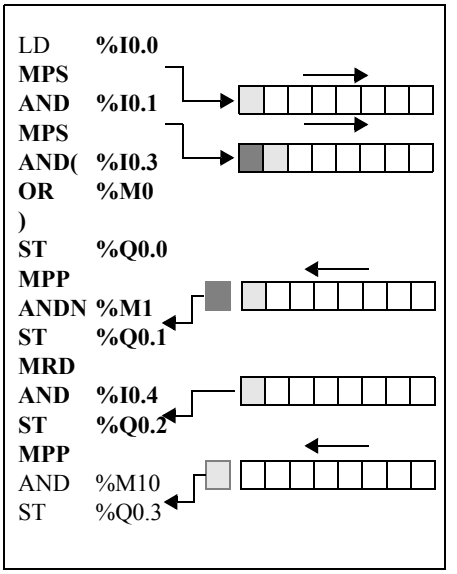



```
LD      %I0.0
AND     %M1
MPS
AND     %I0.1
ST      %Q0.0
MRD
AND     %I0.2
ST      %Q0.1
MRD
AND     %I0.3
ST      %Q0.2
MPP
AND     %I0.4
ST      %Q0.3
```

Exemples du fonctionnement de la pile

Les schémas suivants illustrent le fonctionnement des instructions de pile.





Grafcet

16

Présentation

Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage Grafcet.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description des instructions Grafcet	420
Description de la structure d'un programme Grafcet	425
Actions associées aux étapes Grafcet	429

Description des instructions Grafcet

Introduction

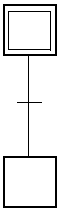
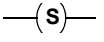
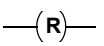
Les instructions Grafcet de TwidoSuite offrent une méthode simple de traduction de séquences de contrôle (graphique Grafcet).

Le nombre maximum d'étapes Grafcet dépend du type d'automate Twido. Le nombre d'étapes pouvant être activées simultanément est uniquement limité par le nombre total d'étapes.

Pour les automates TWDLCAA10DRF et TWDLCAA16DRF, vous disposez des étapes 1 à 62. Les étapes 0 et 63 sont réservées pour le traitement antérieur et postérieur. Pour tous les autres automates, vous disposez des étapes 1 à 95.

Instructions Grafcet

Le tableau suivant répertorie toutes les instructions et les objets requis pour la programmation d'un graphique Grafcet.

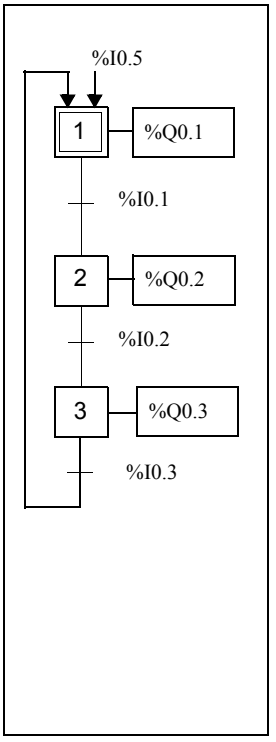
Représentation graphique (1)	Transcription en langage TwidoSuite	Fonction
illustration :  **Etape initiale** **Transition** **Etape**	=*= i	Lance l'étape initiale (2).
	# i	Active l'étape i après avoir désactivé l'étape courante.
	-*- i	Lance l'étape i et valide la transition associée (2).
	#	Désactive l'étape courante sans activer d'autre étape.
	#Di	Désactive l'étape i et l'étape courante.
	=*= POST	Lance le traitement postérieur et termine le traitement séquentiel.
	%Xi	Bit associé à l'étape i. Peut être testé et écrit (le nombre maximum d'étapes dépend de l'automate).
Xi 	LD %Xi, LDN %Xi AND %Xi, ANDN %Xi, OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Teste l'activité de l'étape i.
	S %Xi	Active l'étape i.
Xi  **Xi** 	R %Xi	Désactive l'étape i.

(1) La représentation graphique n'est pas prise en charge.

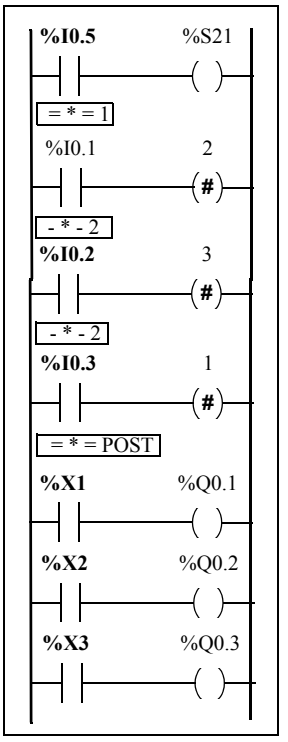
(2) La première étape =*=i ou -*-i écrite indique le lancement du traitement séquentiel et, par conséquent, la fin du prétraitement.

Exemples Grafcet

Séquence linéaire :



Non pris en charge

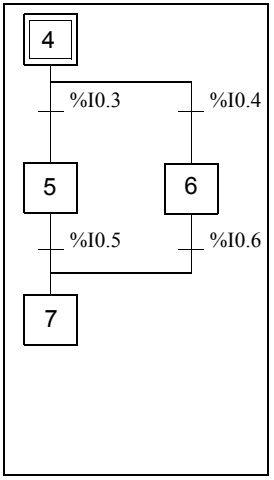


Programme schéma à contacts Twido

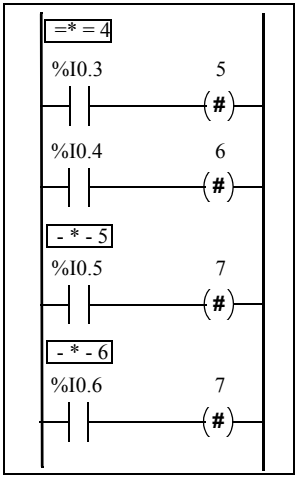
```
LD      %I0.5
ST      %S21
=*=     1
LD      %I0.1
 #      2
-*-     2
LD      %I0.2
 #      3
-*-     3
LD      %I0.3
 #      1
=*=   POST
LD      %X1
ST      %Q0.1
LD      %X2
ST      %Q0.2
LD      %X3
ST      %Q0.3
```

Programme liste d'instructions Twido

Séquence de divergences :



Non pris en charge

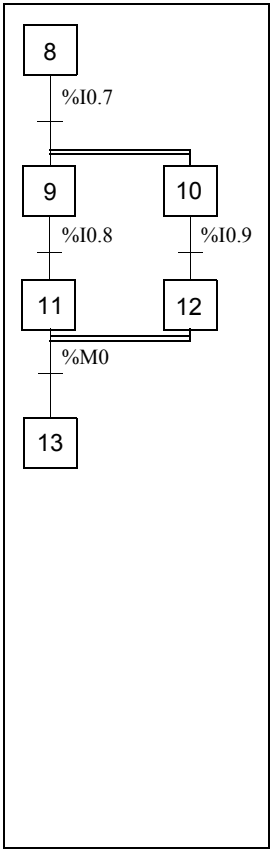


Programme schéma à contacts Twido

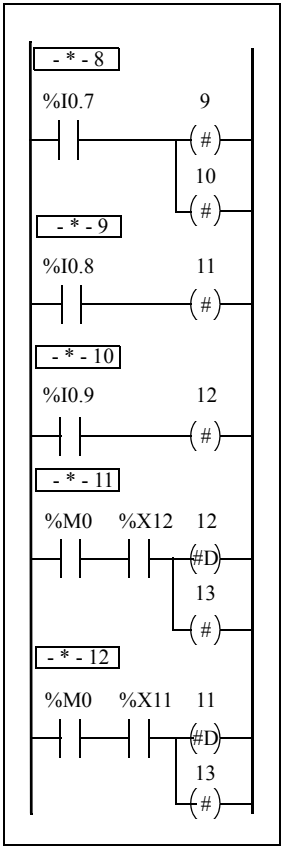
```
=*=     4
 LD     %I0.3
 #      5
 LD     %I0.4
 #      6

 -*-    5
 LD     %I0.5
 #      7

-*-     6
 LD     %I0.6
 #      7
```

Programme liste d'instructions Twido

Séquences simultanées :



Non pris en charge



Programme schéma à contacts Twido

```
 -*-      8
 LD       %I0.7
 #        9
 #        10

-*-       9
 LD       %I0.8
 #        11

-*-       10
 LD       %I0.9
 #        12

-*-       11
 LD       %M0
 AND      %X12
 #D       12
 #        13

-*-       12
 LD       %M0
 AND      %X11
 #D       11
 #        13
```

Programme liste d'instructions Twido

Note : Pour qu'un graphique Grafcet soit opérationnel, au moins une étape active doit être déclarée à l'aide de l'instruction =*=i (étape initiale) ou le graphique doit être prépositionné lors du prétraitement à l'aide du bit système %S23 et de l'instruction S %Xi.

Description de la structure d'un programme Grafcet

Introduction

Un programme TwidoSuite Grafcet se déroule en trois phases :

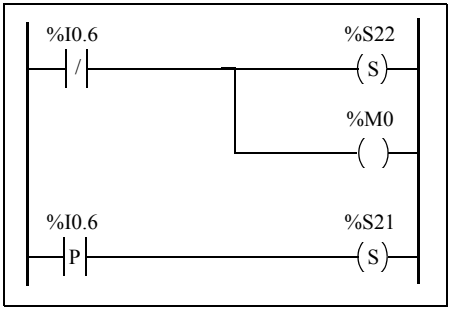
- Pré-traitement
- Traitement séquentiel
- Traitement postérieur

Pré-traitement

Le pré-traitement gère les éléments suivants :

- les reprises de l'alimentation ;
- les défauts ;
- les changements du mode de marche ;
- le pré-positionnement des étapes Grafcet ;
- la logique d'entrée.

Le front montant de l'entrée %I0.6 met à 1 le bit %S21. Cela a pour effet de désactiver les étapes actives et d'activer les étapes initiales.




```
000     LDN     %I0.6
001     S       %S22
002     ST      %M0
003     LDR     %I0.6
004     S       %S21
```

Le pré-traitement commence à la première ligne du programme et se termine à la première occurrence d'une instruction "= * =" ou "- * -".

Trois bits système sont dédiés à la régulation Grafcet : %S21, %S22 et %S23. Chaque bit système est mis à 1 (si nécessaire) par l'application, lors du pré-traitement généralement. La fonction associée est exécutée par le système à la fin du pré-traitement et le bit système est remis à 0 par le système.

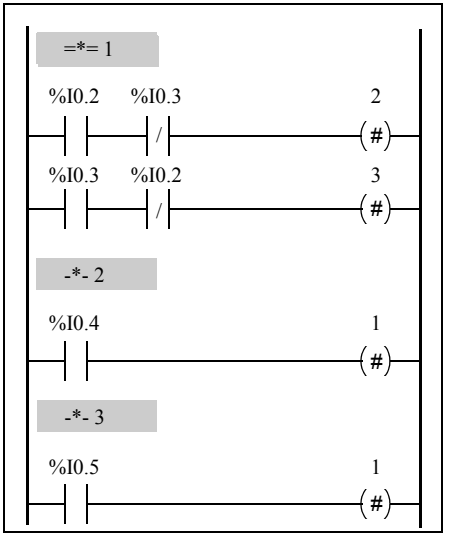
Bit système	Nom	Description
%S21	Initialisation du Grafcet	Toutes les étapes actives sont désactivées et les étapes initiales sont activées.
%S22	Réinitialisation du Grafcet	Toutes les étapes sont désactivées.
%S23	Pré-positionnement du Grafcet	Ce bit doit être mis à 1 si les objets %Xi sont explicitement écrits par l'application lors du pré-traitement. Si ce bit est maintenu sur 1 lors du pré-traitement sans changement explicite des objets %Xi, le Grafcet est figé (aucune mise à jour n'est prise en compte).

Traitement séquentiel

Le traitement séquentiel est exécuté dans le graphique (instructions représentant le graphique) :

- étapes
- actions associées aux étapes
- transitions
- conditions de transition

Exemple :




```
005     =*=     1
006     LD      %I0.2
007     ANDN    %I0.3
008     #       2
009     LD      %I0.3
010     ANDN    %I0.2
011     #       3
012     -*-     2
013     LD      %I0.4
014     #       1
015     -*-     3
016     LD      %I0.5
017     #       1
```

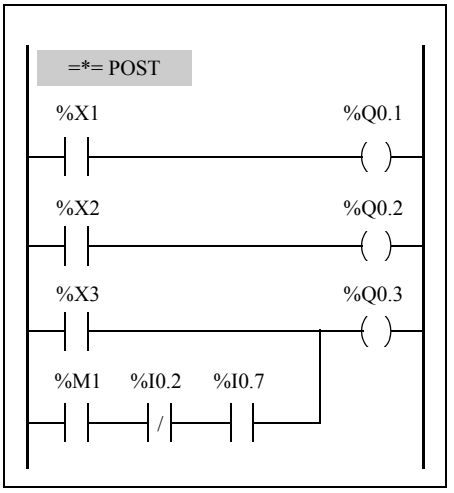
Le traitement séquentiel se termine par l'exécution de l'instruction "= * = POST" ou par la fin du programme.

Traitement postérieur

Le traitement postérieur gère les éléments suivants :

- les commandes du traitement séquentiel pour la régulation des sorties ;
- le verrouillage de sécurité spécifique aux sorties.

Exemple :




```
018     =*=     POST
019     LD      %X1
020     ST      %Q0.1
021     LD      %X2
022     ST      %Q0.2
023     LD      %X3
024     OR(     %M1
025     ANDN    %I0.2
026     AND     %I0.7
027     )
028     ST      %Q0.3
```

Actions associées aux étapes Grafcet

Introduction

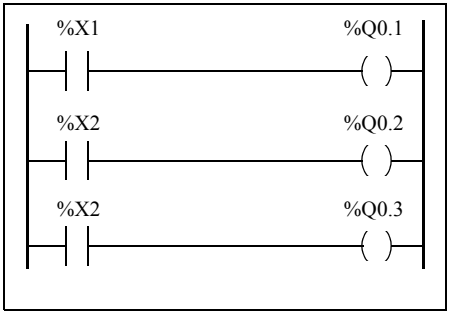
Un programme Grafcet TwidoSuite offre deux modes de programmation des actions associées aux étapes :

- dans la section de traitement postérieur ;
- dans les listes d'instructions ou les réseaux schéma à contacts des étapes mêmes.

Association des actions dans le traitement postérieur

Si des contraintes de sécurité ou de mode d'exécution sont appliquées, il est préférable de programmer les actions dans la section de traitement postérieur d'une application Grafcet. Vous pouvez utiliser les instructions en langage liste d'instructions SET et RESET ou activer les bobines d'un programme schéma à contacts pour lancer les étapes Grafcet (%Xi).

Exemple :

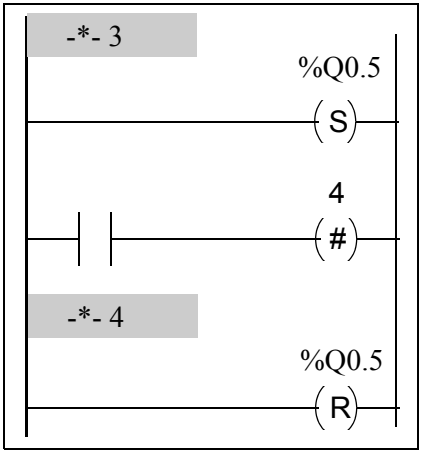



```
018     =*=     POST
019     LD      %X1
020     ST      %Q0.1
021     LD      %X2
022     ST      %Q0.2
023     LD      %X3
024     ST      %Q0.3
```

Association d'actions à partir d'une application

Vous pouvez programmer les actions associées aux étapes au sein de réseaux listes d'instructions ou schéma à contacts. Dans ce cas, le réseau liste d'instructions ou schéma à contacts n'est pas scruté tant que l'étape n'est pas active. Ce mode d'utilisation du langage Grafcet est le plus efficace, le plus lisible et le plus facile à gérer.

Exemple :




```
020   -*-   3
021   LD    1
022   S     %Q0.5
023   LD    %M10
024   #     4
025   -*-   4
026   LD    1
027   R     %Q0.5
028   ...
029   ...
```

Description des instructions et des fonctions

IV

Présentation

Objet de cette partie

Cette partie fournit des descriptions détaillées des instructions élémentaires et avancées, ainsi que des bits et des mots système des langages Twido.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
17	Instructions élémentaires	433
18	Instructions avancées	501
19	Bits système et mots système	673

Instructions élémentaires

17

Présentation

Objet de ce chapitre

Ce chapitre fournit des détails sur les instructions et les blocs fonctions utilisés pour créer des programmes de régulation élémentaires des automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
17.1	Traitement booléen	434
17.2	Blocs fonctions élémentaires	450
17.3	Traitement numérique	474
17.4	Instructions du programme	493

17.1 Traitement booléen

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre offre une introduction au traitement booléen. Elle s'appuie sur des descriptions et des directives de programmation d'instructions booléennes.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions booléennes	435
Explication du format de description des instructions booléennes	437
Instructions de chargement (LD, LDN, LDR, LDF)	439
Instructions d'affectation (ST, STN, R, S)	441
Instructions AND logique (AND, ANDN, ANDR, ANDF)	443
Instructions OR logique (OR, ORN, ORR, ORF)	445
OR exclusif, instructions (XOR, XORN, XORR, XORF)	447
Instruction NOT (N)	449

Instructions booléennes

Introduction

Les instructions booléennes s'apparentent aux éléments graphiques du langage schéma à contacts. Ces instructions sont présentées dans le tableau suivant.

Elément	Instruction	Exemple	Description
Eléments de test	L'instruction de chargement (LD) équivaut à un contact ouvert.	LD %I0.0	Le contact est fermé lorsque le bit %I0.0 se trouve à l'état 1.
Eléments d'action	L'instruction de stockage (ST) équivaut à une bobine.	ST %Q0.0	L'objet bit associé prend la valeur logique de l'accumulateur de bit (résultat de la logique précédente).

Le résultat booléen des éléments de test est appliqué aux éléments d'action, comme l'illustrent les instructions suivantes.

```
LD    %I0.0
AND   %I0.1
ST    %Q0.0
```

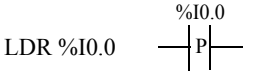
Test des entrées de l'automate

Des instructions sur test booléennes peuvent être utilisées pour détecter des fronts montants ou descendants sur les entrées de l'automate. Un front est détecté lorsque l'état d'une entrée est passé de la valeur "scrutation n-1" à la valeur "scrutation n" courante. La détection de ce front reste effective pendant la scrutation courante.

Front montant, détection

L'instruction LDR (Load Rising Edge - chargement du front montant) équivaut à un contact de détection d'un front montant. Le front montant détecte le passage de la valeur d'entrée de 0 à 1.

Un contact de détection de transition positive est utilisé pour détecter un front montant, comme l'illustre le schéma suivant.



P : contact de détection de transition positive

Détection d'un front descendant

L'instruction LDF (Load Falling Edge - chargement du front descendant) équivaut à un contact de détection d'un front descendant. Le front descendant détecte le passage de la valeur d'entrée de régulation de 1 à 0.

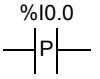
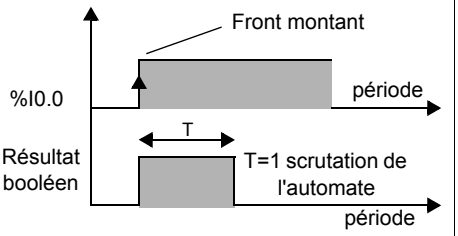
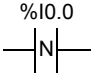
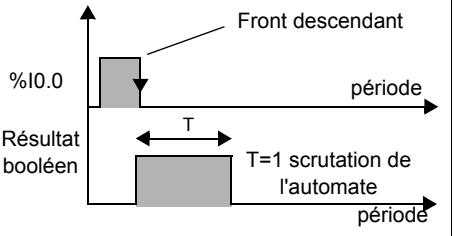
Un contact de détection de transition négative est utilisé pour détecter un front descendant, comme l'illustre le schéma suivant.

LDF %I0.0

N : contact de détection de transition négative

Détection d'un front

Le tableau suivant résume les instructions de détection de fronts et leurs chronogrammes :

Front	Instruction sur test	Schéma à contacts	Chronogramme
Front montant	LDR %I0.0		
Front descendant	LDF %I0.0		

Note : Il est dorénavant possible d'appliquer les instructions sur front aux bits internes %Mi.

Explication du format de description des instructions booléennes

Introduction

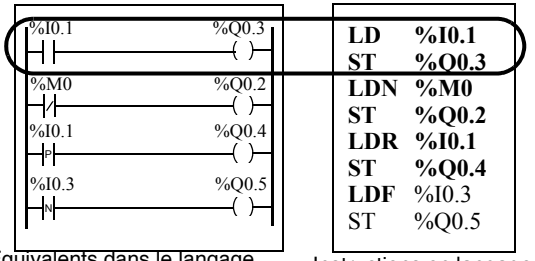
Chaque instruction booléenne de ce sous-chapitre est décrite à l'aide des informations suivantes :

- Description rapide
- Exemple représentant l'instruction et le schéma à contacts correspondant
- Liste d'opérandes autorisés
- Chronogramme

Les explications ci-dessous présentent plus en détails le mode de description des instructions booléennes de ce sous-chapitre.

Exemples

L'illustration suivante présente le mode d'affichage des exemples pour chaque instruction.



Equivalents dans le langage schéma à contacts

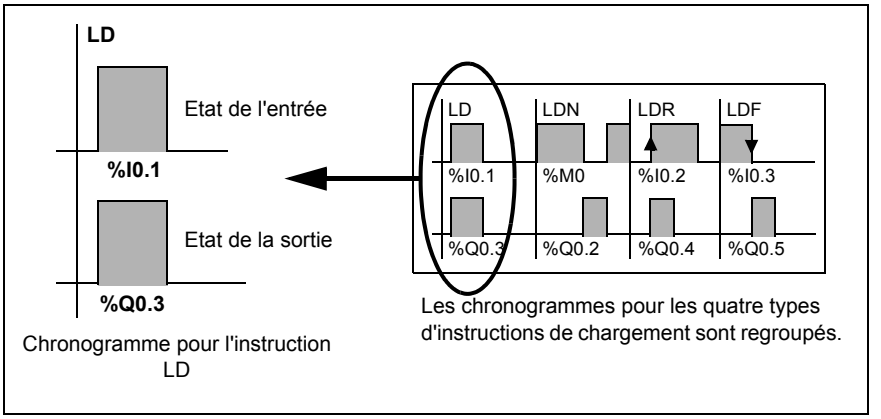
Instructions en langage liste d'instructions

Opérandes autorisés

Le tableau suivant définit les types d'opérandes autorisés et utilisés dans les instructions booléennes.

Opérande	Description
0/1	Valeur immédiate de 0 ou 1
%I	Entrée automate %Ii.j
%Q	Sortie automate %Qi.j
%M	Bit interne %Mi
%S	Bit système %Si
%X	Bit étape %Xi
%BLK.x	Bit bloc fonction (%TMi.Q, par exemple)
%•:Xk	Bit mot (%MWi:Xk, par exemple)
[	Expression de comparaison ([%MWi<1000], par exemple)

Chronogrammes L'illustration suivante présente le mode d'affichage des chronogrammes pour chaque instruction.



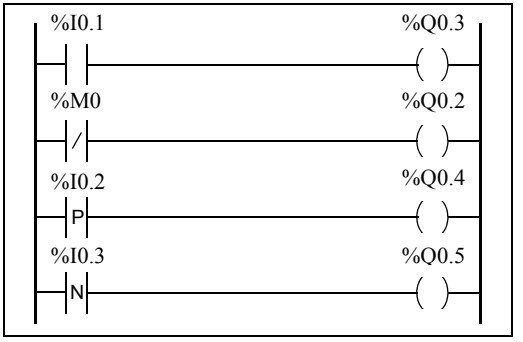
Instructions de chargement (LD, LDN, LDR, LDF)

Introduction

Les instructions de chargement LD, LDN, LDR et LDF correspondent respectivement aux contacts ouverts, fermés, à front montant et à front descendant (les instructions LDR et LDF ne sont utilisées qu'avec des entrées et des mots internes de l'automate et des entrées d'esclaves AS-Interface et PDO CANopen).

Exemples

Les schémas suivants sont des exemples d'instructions de chargement.



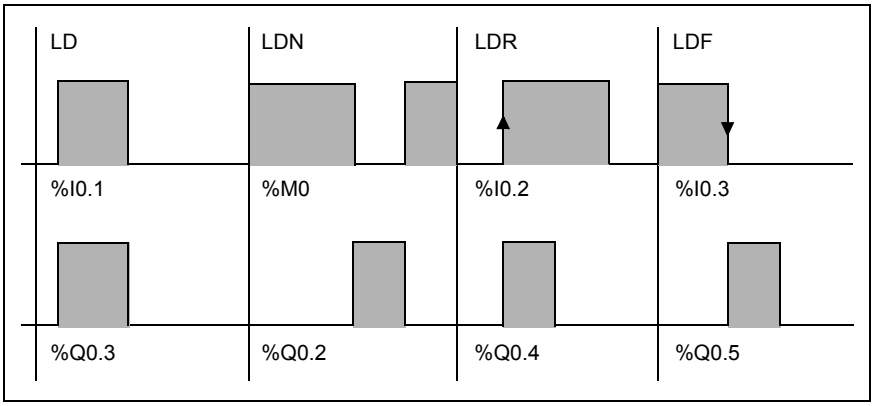

```
LD    %I0.1
ST    %Q0.3
LDN   %M0
ST    %Q0.2
LDR   %I0.2
ST    %Q0.4
LDF   %I0.3
ST    %Q0.5
```

Opérandes autorisés

Le tableau suivant répertorie les types d'instructions de chargement, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
LD		0/1, %I, %IA, %IWCx.y.z:Xk, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk,[
LDN		0/1, %I, %IA, %IWCx.y.z:Xk, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk,[
LDR		%I, %IA, %M
LDF		%I, %IA, %M

Chronogramme L'illustration suivante montre le chronogramme des instructions de chargement :



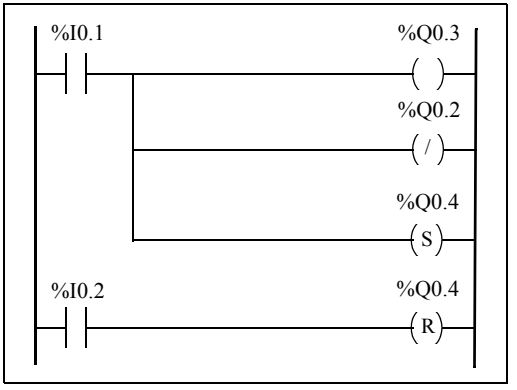
Instructions d'affectation (ST, STN, R, S)

Introduction

Les instructions d'affectation ST, STN, S et R correspondent respectivement aux bobines directes, inverses, d'enclenchement et de déclenchement.

Exemples

Les schémas suivants sont des exemples d'instructions d'affectation.



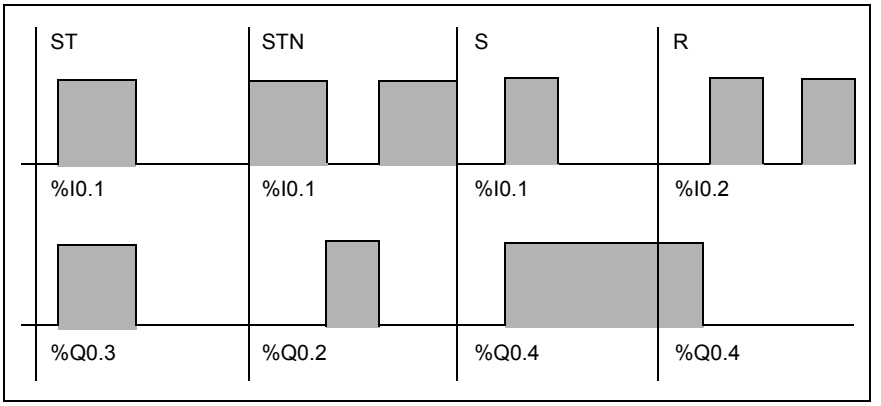

```
LD      %I0.1
ST      %Q0.3

STN     %Q0.2
S       %Q0.4

LD      %I0.2
R       %Q0.4
```

Opérandes autorisés

Le tableau suivant répertorie les types d'instructions d'affectation, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
ST	( )	%Q,%QA,%M,%S,%BLK.x,%•:Xk
STN	(/)	%Q,%QA%M,%S,%BLK.x,%•:Xk
S	(S)	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk
R	(R)	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk

Chronogramme L'illustration suivante montre le chronogramme des instructions d'affectation.



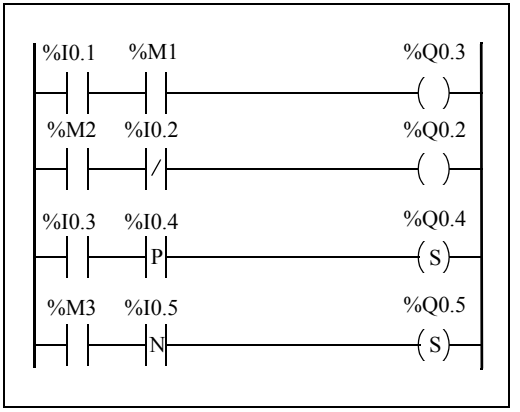
Instructions AND logique (AND, ANDN, ANDR, ANDF)

Introduction

Les instructions AND effectuent une opération de liaison AND logique entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

Exemples

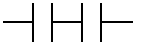
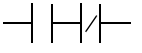
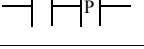
Les schémas suivants sont des exemples d'instructions AND.



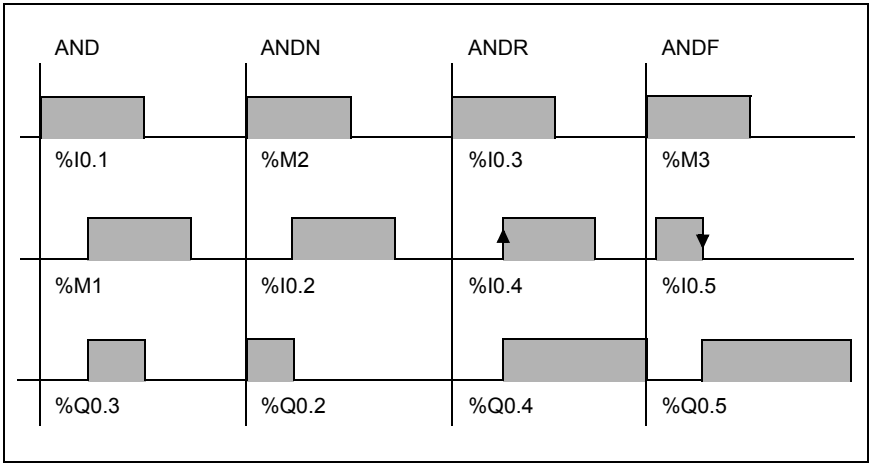

```
LD      %I0.1
AND     %M1
ST      %Q0.3
LD      %M2
ANDN    %I0.2
ST      %Q0.2
LD      %I0.3
ANDR    %I0.4
S       %Q0.4
LD      %M3
ANDF    %I0.5
S       %Q0.5
```

Opérandes autorisés

Le tableau suivant répertorie les types d'instructions AND, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
AND		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDN		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDR		%I, %IA, %M
ANDF		%I, %IA, %M

Chronogramme Le diagramme suivant illustre la temporisation des instructions AND.



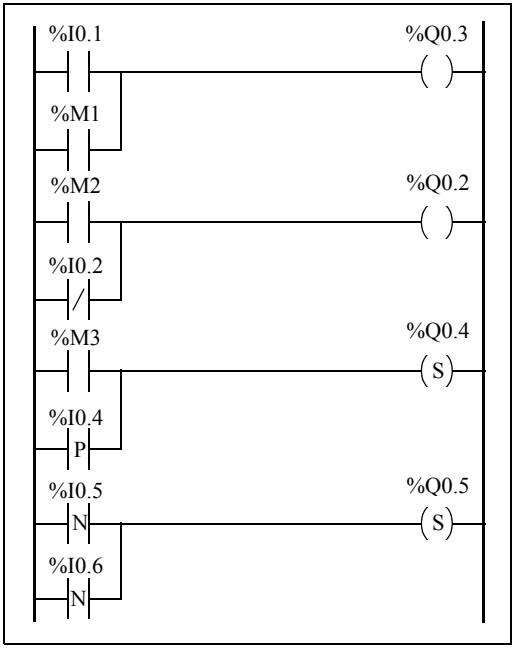
Instructions OR logique (OR, ORN, ORR, ORF)

Introduction

Les instructions OR effectuent une opération de liaison OR logique entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

Exemples

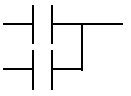
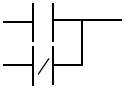
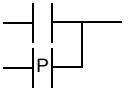
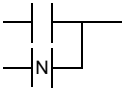
Les schémas suivants sont des exemples d'instructions OR.




```
LD      %I0.1
OR      %M1
ST      %Q0.3

LD      %M2
ORN     %I0.2
ST      %Q0.2

LD      %M3
ORR     %I0.4
S       %Q0.4

LDF     %I0.5
ORF     %I0.6
S       %Q0.5
```

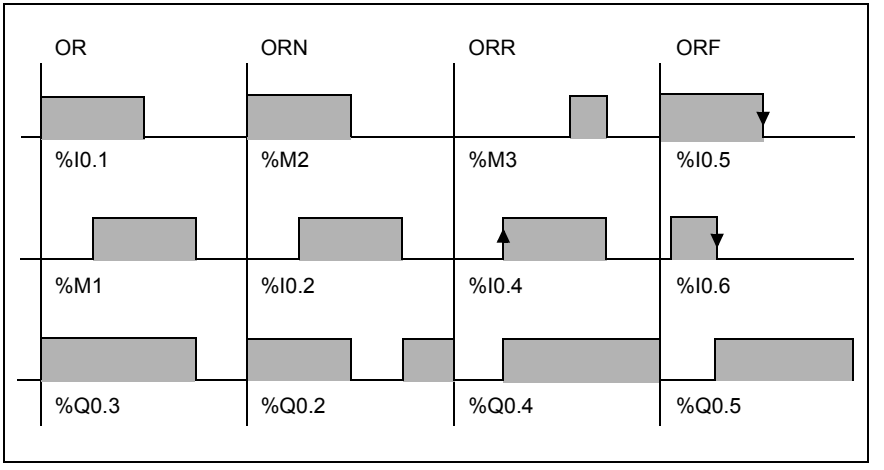
Opérandes autorisés

Le tableau suivant répertorie les types d'instructions OR, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
OR		0/1, %I,%IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORN		0/1, %I,%IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORR		%I, %IA, %M
ORF		%I, %IA, %M

Chronogramme

L'illustration suivante montre le chronogramme des instructions OR.



OR exclusif, instructions (XOR, XORN, XORR, XORF)

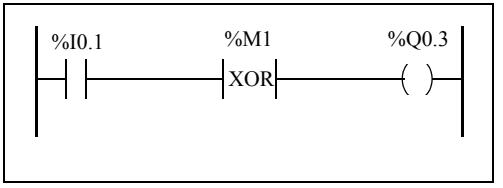
Introduction

Les instructions XOR effectuent une opération de liaison OR exclusif entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

Exemples

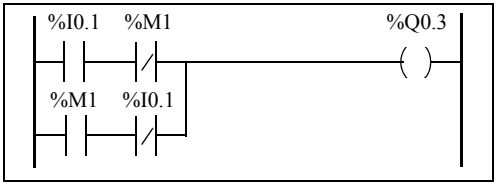
L'exemple suivant illustre l'utilisation d'instructions XOR.

Schéma utilisant des instructions XOR :




```
LD      %I0.1
XOR     %M1
ST      %Q0.3
```

Schéma n'utilisant PAS d'instructions XOR :




```
LD      %I0.1
ANDN    %M1
OR(     %M1
ANDN    %I0.1
)
ST      %Q0.3
```

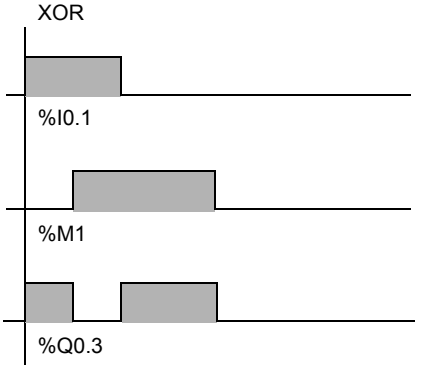
Opérandes autorisées

Le tableau suivant répertorie les types d'instructions XOR, ainsi que les opérandes autorisés.

Instruction langage liste	Opérandes autorisés
XOR	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
XORN	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
XORR	%I, %IA, %M
XORF	%I, %IA, %M

Chronogramme

Le diagramme suivant illustre la temporisation des instructions XOR.

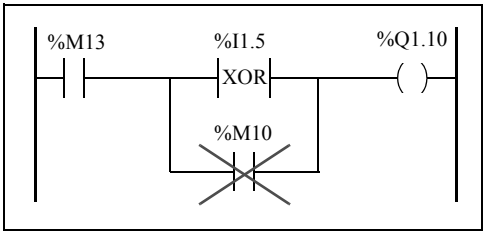


Cas spécifiques

Veuillez observer les précautions suivantes lors de l'utilisation d'instructions XOR dans des programmes en langage schéma à contacts :

- Ne commencez jamais un réseau par un contact XOR.
- N'insérez jamais de contacts XOR parallèlement à d'autres éléments du schéma à contacts (reportez-vous à l'exemple suivant.)

Comme l'illustre l'exemple suivant, l'insertion d'un élément parallèle à un contact XOR générera une erreur de validation.



Instruction NOT (N)

Introduction L'instruction NOT (N) inverse le résultat booléen de l'instruction précédente.

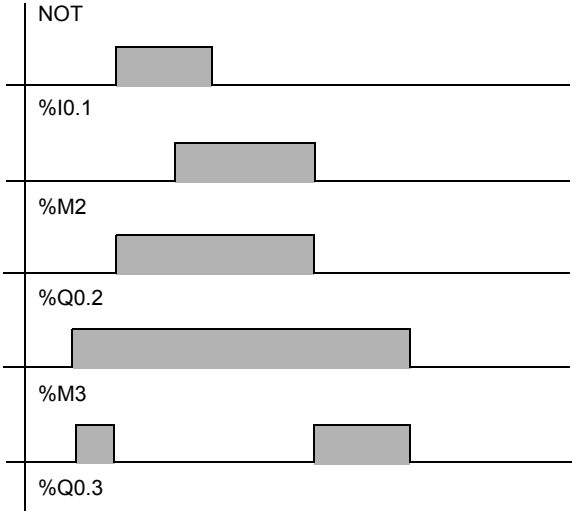
Exemple L'exemple suivant illustre un exemple de l'utilisation de l'instruction NOT.

```
LD      %I0.1
OR      %M2
ST      %Q0.2
N
AND     %M3
ST      %Q0.3
```

Note : L'instruction NOT n'est pas réversible.

Opérandes autorisées Sans objet.

Chronogramme L'illustration suivante montre le chronogramme de l'instruction NOT.



17.2 Blocs fonctions élémentaires

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre présente des descriptions et des conseils de programmation relatifs aux blocs fonctions élémentaires.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Blocs fonctions standard	451
Principes de programmation de blocs fonction standard	453
Bloc fonction temporisateur (%TMi)	455
Type de temporisateur TOF	457
Type de temporisateur TON	458
Type de temporisateur TP	459
Programmation et configuration de temporisateurs	460
Bloc fonction compteur/décompteur (%Ci)	463
Programmation et configuration des compteurs	467
Bloc fonction registre bits à décalage (%SBRi)	470
Bloc fonction pas à pas (%SCi)	472

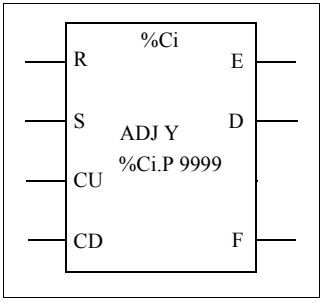
Blocs fonctions standard

Introduction

Les blocs fonctions sont les sources des objets bits et des mots spécifiques utilisés par les programmes. Les blocs fonctions élémentaires comportent des fonctions simples telles que des temporisateurs ou des compteurs/décompteurs.

Exemple de bloc fonction

L'illustration suivante présente un exemple de bloc fonction compteur/décompteur.



Bloc compteur/décompteur

Objets bits

Les objets bits correspondent aux sorties des blocs. Les instructions booléennes de test peuvent accéder à ces bits selon l'une ou l'autre de ces méthodes :

- directement (LD E, par exemple) s'ils sont liés au bloc par une programmation réversible (voir *p. 453*).
- en spécifiant le type de bloc (LD %Ci.E, par exemple).

Les entrées sont accessibles sous forme d'instructions.

Objets mots

Les objets mots correspondent aux :

- **Paramètres de configuration des blocs :** le programme peut accéder à certains paramètres (paramètres de présélection, par exemple), mais pas à d'autres (base temps, par exemple).
- **Valeurs courantes :** %Ci.V, la valeur de comptage courante, par exemple.

Objets bits et objets mots accessibles

Le tableau suivant décrit les objets bits et les objets mots de blocs fonctions auxquels le programme a accès.

Bloc fonction standard	Symbole	Plage (i)	Types d'objets	Description	Adresse	Accès en mode écriture
Temporisateur	%TMi	0 - 127	Mot	Valeur courante	%TMi.V	non
				Valeur de présélection	%TMi.P	oui
			Bit	Sortie du temporisateur	%TMi.Q	non
Compteur/ Décompteur	%Ci	0 - 127	Mot	Valeur courante	%Ci.V	non
				Valeur de présélection	%Ci.P	oui
			Bit	Sortie pour dépassement par valeur inférieure (vide)	%Ci.E	non
				Sortie prédéfinie atteinte	%Ci.D	non
				Sortie pour débordement (plein)	%Ci.F	non

Principes de programmation de blocs fonction standard

Introduction

Pour programmer des blocs fonction standard, appliquez l'une des méthodes suivantes :

- Instructions sur bloc fonction (par exemple `BLK` %TM2) : Cette méthode de programmation en langage schéma à contacts réversible permet l'exécution d'opérations sur le bloc, à un emplacement unique du programme.
- Instructions spécifiques (par exemple `CU %Ci`) : Cette méthode non réversible permet l'exécution d'opérations sur les entrées du bloc, à plusieurs emplacements du programme (par exemple, `line 100 CU %C1`, `line 174 CD %C1`, `line 209 LD %C1.D`).

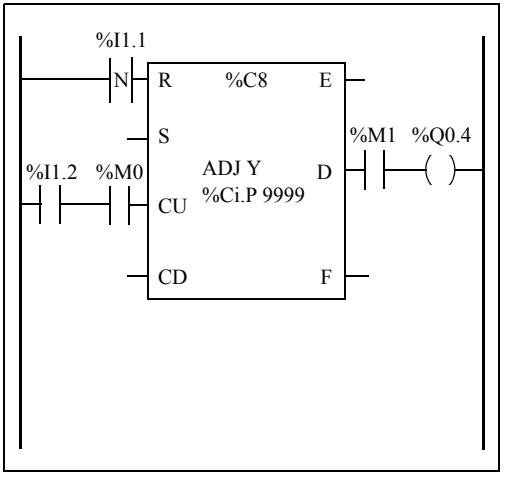
Programmation réversible

Utilisez les instructions BLK, OUT_BLK et END_BLK pour une programmation réversible :

- **BLK :** Indique le début du bloc.
- **OUT_BLK :** Utilisé pour câbler directement les sorties du bloc.
- **END_BLK :** Indique la fin du bloc.

Exemple avec des sorties câblées

Vous trouverez ci-dessous un exemple de programmation réversible d'un bloc fonction compteur avec des sorties câblées.



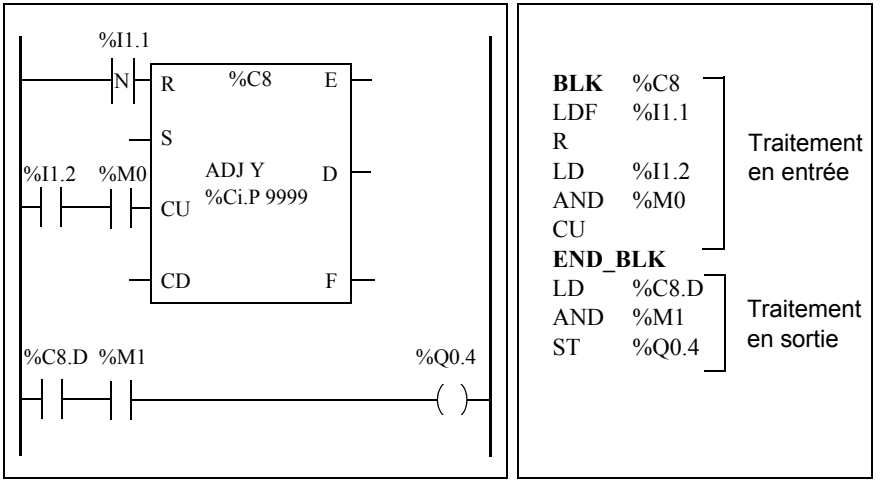

```
BLK     %C8
LDF     %I1.1
R
LD      %I1.2
AND     %M0
CU
OUT_BLK
LD      D
AND     %M1
ST      %Q0.4
END_BLK
```

Traitement en entrée

Traitement en sortie

Exemple sans sortie câblée

Vous trouverez ci-dessous un exemple de programmation réversible d'un bloc fonction compteur dépourvu de sortie câblée.



Note : Seules les instructions de test et d'entrée sur le bloc correspondant peuvent être placées entre les instructions BLK et OUT_BLK (ou entre BLK et END_BLK lorsque OUT_BLK n'est pas programmé).

Bloc fonction temporisateur (%TMi)

Introduction

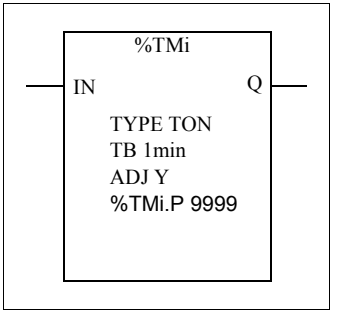
Il existe trois types de blocs fonction temporisateur :

- TON (temporisateur de retard à l'enclenchement) : ce type de temporisateur permet de gérer les retards à l'enclenchement.
- TOF (temporisateur de retard au déclenchement) : ce type de temporisateur permet de gérer les retards au déclenchement.
- TP (temporisateur - Impulsion) : ce type de temporisateur permet de générer des impulsions d'une durée précise.

TwidoSuite permet de programmer et de modifier les retards de ces temporisateurs et/ou les durées des impulsions qu'ils génèrent.

Illustration

L'exemple suivant illustre l'utilisation du bloc fonction temporisateur.



Bloc fonction temporisateur

Paramètres

Le bloc fonction temporisateur possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro du temporisateur	%TMi	0 à 63 : TWDLCAA10DRF et TWDLCAA16DRF 0 à 127 pour tous les autres automates.
Type	TON	• retard à l'enclenchement (par défaut)
	TOF	• retard au déclenchement
	TP	• impulsion (monostable)
Base de temps	TB	1 min (par défaut), 1 s, 100 ms, 10 ms, 1 ms
Valeur courante	%TMi.V	Mot avec des incréments allant de 0 à %TMi.P lorsque le temporisateur est en cours d'exécution. Peut être lu et testé, mais pas écrit par le programme. %TMi.V peut être modifié par l'éditeur de tables d'animation.
Valeur de présélection	%TMi.P	0 - 9999. Mot pouvant être lu, testé et écrit par le programme. La valeur par défaut est 9999. La période ou le délai généré est égal à %TMi.P x TB.
Editeur de tables d'animation	O/N	Y : Oui, la valeur %TMi.P de présélection peut être modifiée à l'aide de l'éditeur de tables d'animation. N : Non, la valeur %TMi.P de présélection ne peut pas être modifiée.
Entrée validation (ou de l'instruction)	IN	Démarre le temporisateur sur le front montant (types TON ou TP) ou descendant (type TOF).
Sortie du temporisateur	Q	Le bit associé %TMi.Q est réglé sur 1 en fonction de la fonction exécutée : TON, TOF ou TP

Note : Plus la valeur de présélection est grande, plus le temporisateur sera précis.

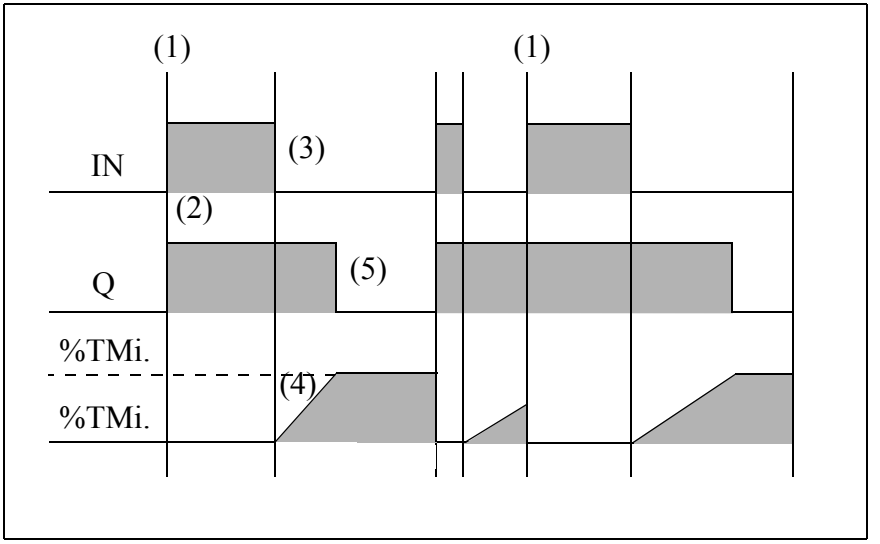
Type de temporisateur TOF

Introduction

Le type de temporisateur TOF (Timer Off-Delay, temporisateur à retard de déclenchement) permet de gérer des retards au déclenchement. TwidoSuite permet de programmer ce retard.

Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TOF.



Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TOF.

Phase	Description
1	La valeur courante %TMi.V prend la valeur 0 sur un front montant en entrée IN, et ce, même si le temporisateur est en cours d'exécution.
2	Le bit de sortie %TMi.Q passe à 1 lorsqu'un front montant est détecté en entrée IN.
3	Le temporisateur démarre sur le front descendant de l'entrée IN.
4	La valeur courante %TMi.V augmente jusqu'à %TMi.P, par incréments d'une unité à chaque pulsation de la base temps TB.
5	Le bit de sortie %TMi.Q est remis à 0 lorsque la valeur courante atteint %TMi.P.

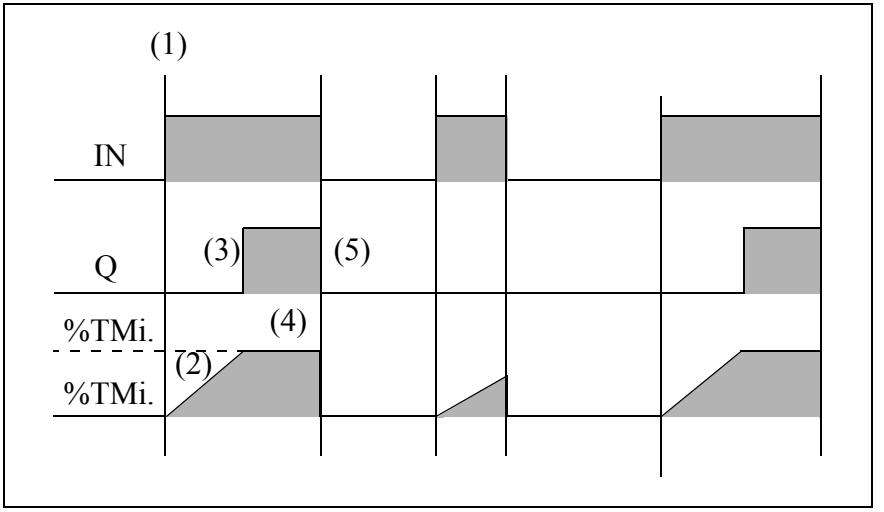
Type de temporisateur TON

Introduction

TON (Timer On-Delay, temporisateur de délai à l'activation) : ce type de temporisateur permet de réguler les actions de délai à l'activation. TwidoSuite permet de programmer cette durée.

Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TON.



Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TON.

Phase	Description
1	Le temporisateur démarre sur le front montant de l'entrée IN.
2	La valeur courante %TMi.V augmente de 0 à %TMi.P, par incréments d'une unité à chaque pulsation de la base temps TB.
3	Le bit de sortie %TMi.Q passe à 1 lorsque la valeur courante a atteint %TMi.P.
4	Le bit de sortie %TMi.Q conserve la valeur 1 tant que la valeur de l'entrée IN est à 1.
5	Lorsqu'un front descendant est détecté en entrée IN, le temporisateur s'arrête, et ce, même s'il n'a pas atteint %TMi.P et que %TMi.V est réglé sur 0.

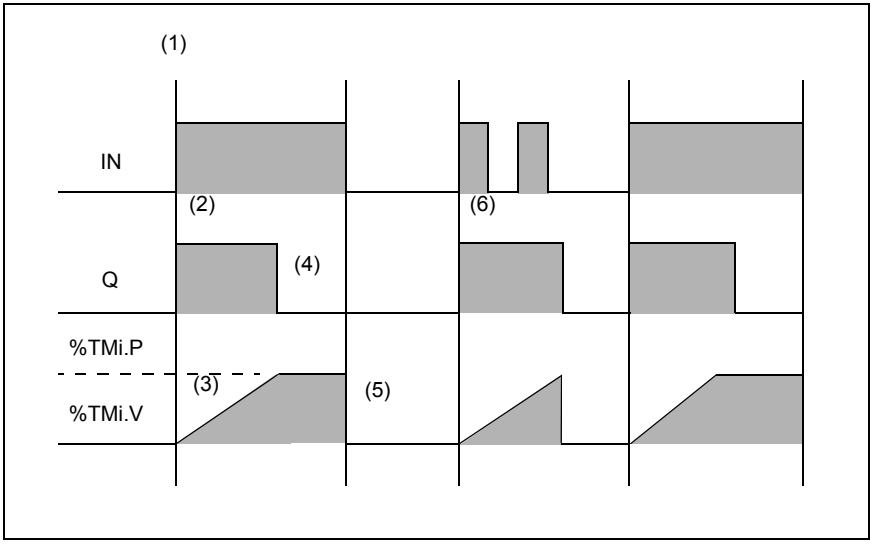
Type de temporisateur TP

Introduction

Le type de temporisateur TP (Timer – Pulse, Temporisateur – Impulsion) permet de générer des impulsions d'une durée spécifique. TwidoSuite permet de programmer cette durée.

Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TP.



Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TP.

Phase	Description
1	Le temporisateur démarre sur le front montant de l'entrée IN. La valeur courante %TMi.V est mis à 0 si le temporisateur n'a pas encore démarré.
2	Le bit de sortie %TMi.Q est mis à 1 lorsque le temporisateur démarre.
3	La valeur courante %TMi.V du temporisateur augmente de 0 à %TMi.P, par incréments d'une unité à chaque pulsation de la base temps TB.
4	Le bit de sortie %TMi.Q est mis à 0 lorsque la valeur courante atteint %TMi.P.
5	La valeur courante %TMi.V est mis à 0 lorsque %TMi.V égale %TMi.P et que l'entrée IN retrouve la valeur 0.
6	Le temporisateur ne peut pas être remis à zéro. Lorsque %TMi.V égale %TMi.P et que l'entrée IN est mis à 0, %TMi.V est réglé sur 0.

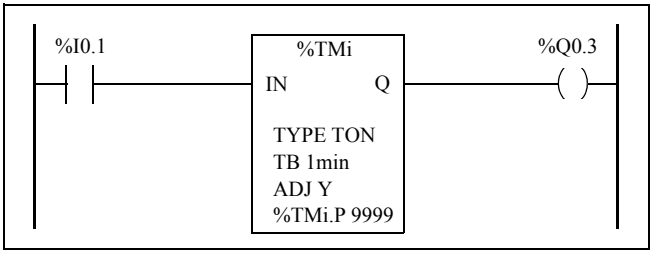
Programmation et configuration de temporisateurs

Introduction

Tous les blocs fonction temporisateur (%TMi) sont programmés de la même façon, indépendamment de leur mode d'utilisation. La fonction temporisateur (TON, TOF ou TP) est sélectionnée au moment de la configuration.

Exemples

L'illustration suivante représente un bloc fonction temporisateur et affiche des exemples de programmation réversible et non réversible.



Programmation réversible

```
BLK     %TM1
LD      %I0.1
IN
OUT_BLK
LD      Q
ST      %Q0.3
END_BLK
```

Programmation non réversible

```
LD      %I0.1
IN      %TM1
LD      %TM1.Q
ST      %Q0.3
```

Configuration

Les paramètres suivants doivent être saisis au moment de la configuration :

- Type de temporisateur : TON, TOF ou TP
- Base temps (TB) : 1 min, 1 s, 100 ms, 10 ms ou 1 ms
- Valeur de présélection (%TMi.P) : 0 à 9999
- Réglage : coché ou non coché

Cas particuliers

Le tableau suivant présente une liste des cas spécifiques de programmation du bloc fonction temporisateur.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Impose 0 à la valeur courante. Règle la sortie %TMi.Q sur 0. La valeur de présélection reprend la valeur réglée au moment de la configuration.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur la valeur courante et la valeur de présélection du temporisateur. La valeur courante n'est pas modifiée lors d'une coupure d'alimentation secteur.
Effet d'un arrêt de l'automate	L'arrêt de l'automate ne provoque pas le gel de la valeur courante.
Effet d'un saut de programme	Le saut d'un bloc temporisateur ne provoque pas le gel du temporisateur. L'incrémentation du temporisateur se poursuit jusqu'à ce que la valeur de présélection (%TMi.P) soit atteinte. A ce stade, l'état du bit Terminé (%TMi.Q) affecté à la sortie Q du bloc temporisateur est modifié. Cependant, la sortie associée, liée directement à la sortie du bloc, n'est ni activée, ni scrutée par l'automate.
Test par bit %TMi.Q (bit terminé)	Nous conseillons de ne tester le bit %TMi.Q qu'une seule fois dans le programme.
Effet de la modification de la valeur de présélection de %TMi.P	La modification de la valeur de présélection à l'aide d'une instruction ou d'un réglage ne prend effet qu'à la prochaine activation du temporisateur.

Temporisateurs avec base temps de 1 ms

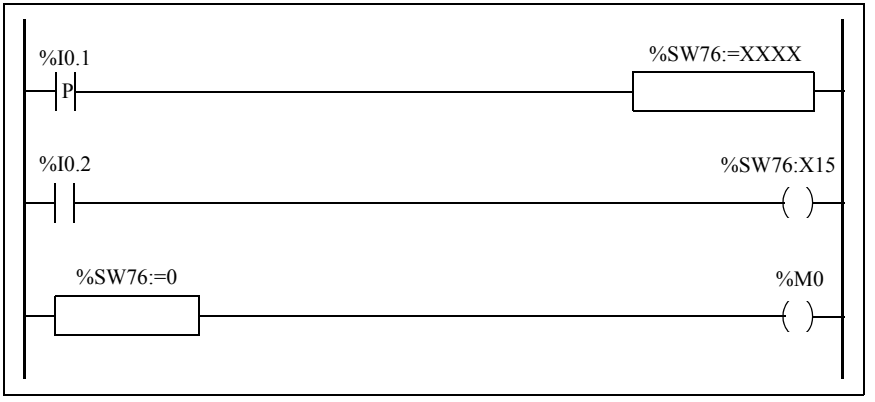
La base temps de 1 ms n'est disponible qu'avec les cinq premiers temporisateurs. Les quatre mots système %SW76, %SW77, %SW78 et SW79 peuvent être utilisés comme des "sabliers". Ces quatre mots sont décrémentés de manière individuelle par le système toutes les millisecondes, **si leur valeur est positive.**

Il est possible de créer une temporisation multiple en chargeant successivement un de ces mots ou en testant les valeurs intermédiaires. Les valeurs négatives de ces quatre mots ne seront pas modifiées. Un temporisateur peut être "gelé" en réglant le bit 15 sur la valeur 1, puis "dégelé" en remettant à zéro cette valeur.

Exemple de programmation

L'exemple suivant illustre la programmation d'un bloc fonction temporisateur.

```
LDR     %I0.1       (Lancement du temporisateur sur le front montant de %I0.1)
[%SW76:=XXXX]  (XXXX = valeur requise)
LD      %I0.2       (gestion optionnelle du gel, gel de l'entrée I0.2)
ST      %SW76:X15
LD      [%SW76=0](test du temporisateur de fin)
ST      %M0
.............
```



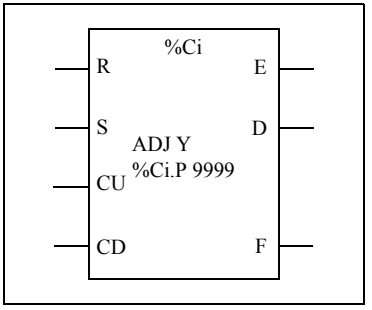
Bloc fonction compteur/décompteur (%Ci)

Introduction

Le bloc fonction compteur (%Ci) permet de compter ou de décompter des événements. Ces deux opérations peuvent être réalisées simultanément.

Illustration

L'illustration suivante présente un exemple de bloc fonction compteur/décompteur.



Bloc fonction compteur/décompteur

Paramètres Le bloc fonction compteur possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro du compteur	%Ci	0 à 127
Valeur courante	%Ci.V	La valeur du mot est augmentée ou diminuée d'une unité en fonction des entrées (ou des instructions) CU et CD. Peut être lue et testée, mais pas écrite par le programme. Utilisez l'Editeur de données pour modifier %Ci.V.
Valeur de présélection	%Ci.P	0 ≤ %Ci.P ≤9 999. Le mot peut être lu, testé et écrit (valeur par défaut : 9999).
Edition à l'aide de l'Editeur de tables d'animation	ADJ	• Y : Oui, la valeur de présélection peut être modifiée à l'aide de l'éditeur de tables d'animation. • N : Non, la valeur de présélection ne peut pas être modifiée à l'aide de l'éditeur de tables d'animation.
Entrée (ou instruction) RAZ	R	A l'état 1 : %Ci.V = 0.
Entrée (ou instruction) de présélection	S	A l'état 1 : %Ci.V = %Ci.P.
Entrée (ou instruction) de comptage	CU	Augmente la valeur de %Ci.V d'une unité sur un front montant.
Entrée (ou instruction) de décomptage	CD	Diminue la valeur de %Ci.V d'une unité sur un front montant.
Sortie débordement décomptage	E (vide)	Le bit associé %Ci.E est égal à 1, lorsque la valeur du décompteur %Ci.V passe de 0 à 9999 (mis à 1 lorsque %Ci.V atteint 9999 et remis à zéro si le décomptage se poursuit).
Sortie prédéfinie atteinte	D (Terminé)	Le bit associé %Ci.D est égal à 1, lorsque %Ci.V est égal à %Ci.P.
Sortie débordement comptage	F (plein)	Le bit associé %Ci.F est égal à 1, lorsque la valeur de %Ci.V passe de 9999 à 0 (mis à 1 lorsque %Ci.V atteint 0 et remis à zéro si le comptage croissant se poursuit).

Fonctionnement Le tableau suivant décrit les étapes principales des opérations de comptage et de décomptage.

Fonctionnement	Action	Résultat
Comptage	Un front montant apparaît sur l'entrée comptage CU (ou l'instruction CU est activée).	La valeur courante de %Ci.V est augmentée d'une unité.
	La valeur courante de %Ci.V est égale à la valeur de présélection de %Ci.P.	Le bit de sortie "présélection atteinte" %Ci.D passe à 1.
	La valeur courante de %Ci.V passe de 9999 à 0.	Le bit de sortie %Ci.F (débordement comptage) passe à 1.
	Si le comptage se poursuit.	Le bit de sortie %Ci.F (débordement comptage) est remis à zéro.
Décomptage	Un front montant apparaît sur l'entrée décomptage CD (ou l'instruction CD est activée).	La valeur courante de %Ci.V est diminuée d'une unité.
	La valeur courante de %Ci.V passe de 0 à 9999.	Le bit de sortie %Ci.E (débordement décomptage) passe à 1.
	Si le décomptage se poursuit.	Le bit de sortie %Ci.F (débordement décomptage) est remis à zéro.
Comptage/ Décomptage	Pour utiliser simultanément les fonctions de comptage et de décomptage (ou pour activer les deux instructions CD et CU), les deux entrées CU et CD correspondantes doivent être commandées simultanément. Ces deux entrées sont ensuite scrutées. Si leur valeur est égale à 1, la valeur courante n'est pas modifiée.	
Remise à zéro	Mise à 1 de l'entrée R (ou l'instruction R est activée).	Force la remise à zéro de la valeur %Ci.V. Les sorties %Ci.E, %Ci.D et %Ci.F sont réglées sur 0. L'entrée remise à zéro est prioritaire.
Présélection	Si l'entrée S est mis à 1 (ou si l'instruction S est activée) et que l'entrée de remise à zéro est à l'état 0 (ou que l'instruction R est inactive).	La valeur courante %Ci.V prend la valeur de %Ci.P et la sortie %Ci.D est mis à 1.

Cas spécifiques Le tableau suivant présente une liste de cas spécifiques de fonctionnement et de configuration des compteurs.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	• La valeur courante de %Ci.V est mise à 0. • Les bits de sortie %Ci.E, %Ci.D et %Ci.F sont mis à sur 0. • La valeur de présélection est initialisée avec la valeur définie au moment de la configuration
Effet d'une reprise à chaud (%S1=1) d'un arrêt de l'automate (STOP)	N'a aucun effet sur la valeur courante du compteur (%Ci.V).
Effet de la modification de la valeur de présélection de %Ci.P	La modification de la valeur de présélection à l'aide d'une instruction ou d'un réglage ne prend effet qu'au moment du traitement du bloc par l'application (activation de l'une des entrées).

Programmation et configuration des compteurs

Introduction

L'exemple suivant illustre un compteur permettant de compter un maximum de 5000 articles. Chaque impulsion sur l'entrée %I1.2 (lorsque le bit interne %M0 est mis à 1) incrémente la valeur du compteur %C8 d'une unité, jusqu'à la valeur de présélection finale (bit %C8.D=1). Le compteur est remis à zéro par l'entrée %I1.1.

Exemple de programmation

L'illustration suivante représente un bloc fonction compteur et affiche des exemples de programmation réversible et non réversible.

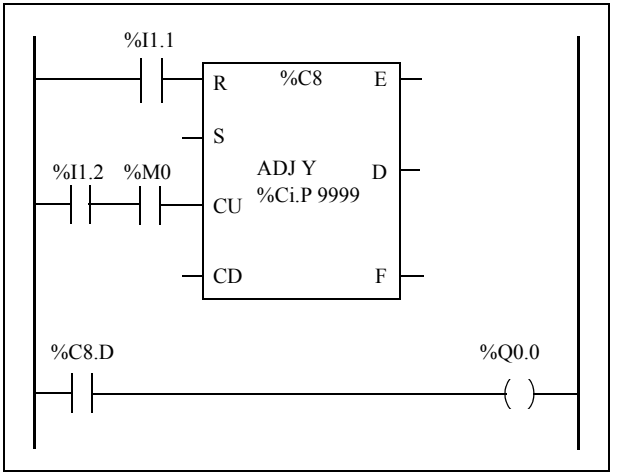


Schéma à contacts

```
BLK     %C8
LD      %I1.1
R
LD      %I1.2
AND     %M0
CU
END_BLK
LD      %C8.D
ST      %Q0.0
```

Programmation réversible

```
LD      %I1.1
R       %C8
LD      %I1.2
AND     %M0
CU      %C8
LD      %C8.D
ST      %Q0.0
```

Programmation non réversible

Configuration

Les paramètres suivants doivent être saisis au moment de la configuration :

- Valeur de présélection (%Ci.P) : fixée à 5000 dans cet exemple
- Réglage : Oui

Exemple d'un compteur/ Décompteur

L'illustration suivante représente un bloc fonction compteur / décompteur.

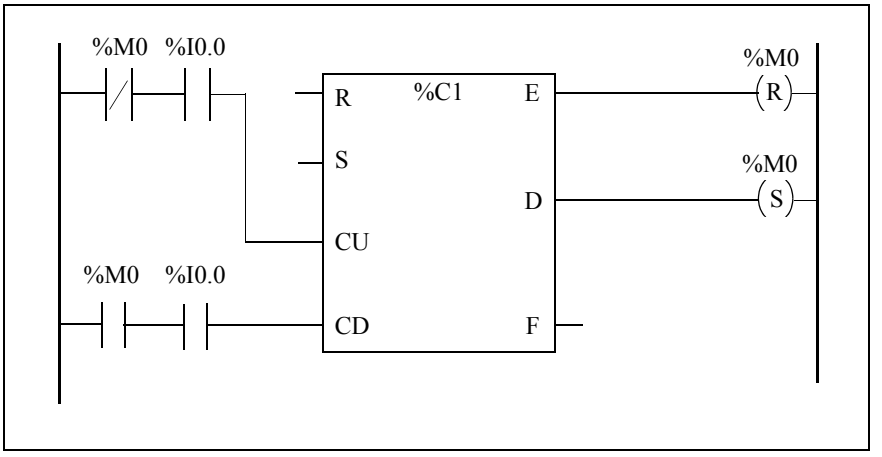


Schéma à contacts

Dans cet exemple, si on prends %C1.P 4, la valeur courante du compteur %C1.V sera incrémenté de 0 jusqu'à 3 puis décrémenté de 3 jusqu'à 0.Tant que %I0.0=1 %C1.V oscille entre 0 et 3.

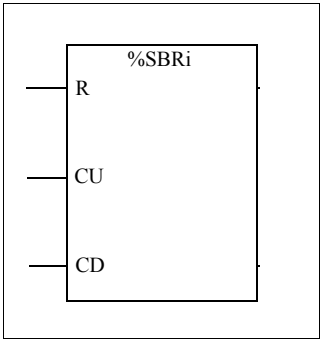
Bloc fonction registre bits à décalage (%SBRi)

Introduction

Le bloc fonction registre bits à décalage (%SBRi) effectue un décalage vers la gauche ou vers la droite des bits de données binaires (0 ou 1).

Illustration

L'exemple suivant illustre un bloc fonction registre à décalage :



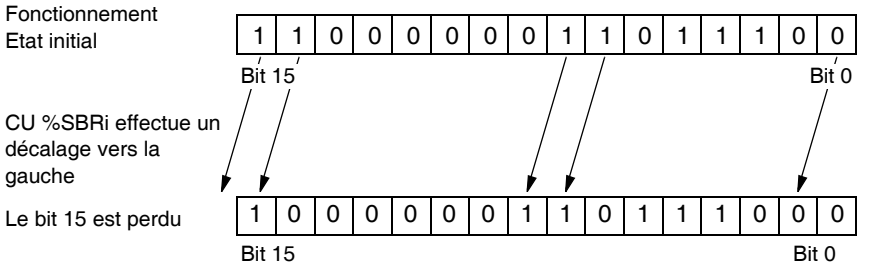
Paramètres

Le bloc fonction registre bits à décalage possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro de registre	%SBRi	0 à 7
Bit de registre	%SBRi.j	Les bits 0 à 15 (j = 0 à 15) du registre à décalage peut être testé par une instruction de test et écrit à l'aide d'une instruction d'affectation.
Entrée (ou instruction) de présélection	R	Lorsque le paramètre fonction R est 1, ceci définit les bits de registre 0 à 15 %SBRi.j sur 0.
Entrée (ou l'instruction) décalage à gauche	CU	Sur un front montant, décale un bit du registre vers la gauche.
Entrée (ou l'instruction) décalage à droite	CD	Sur un front montant, décale un bit du registre vers la droite.

Fonctionnement

L'illustration suivante présente une configuration binaire avant et après une opération de décalage.

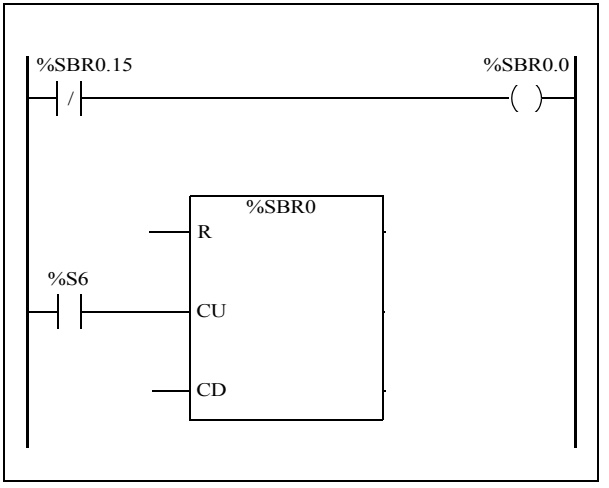

Fonctionnement
Etat initial
1 1 0 0 0 0 0 0 1 1 0 1 1 1 0 0
Bit 15
Bit 0
CU %SBRi effectue un décalage vers la gauche
Le bit 15 est perdu
1 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0
Bit 15
Bit 0

Cet exemple peut également s'appliquer à une requête de décalage d'un bit vers la droite (Bit 15 à Bit 0) à l'aide de l'instruction CD. Le bit 0 est perdu.

Si un registre de 16 bits n'est pas adapté, il est possible d'utiliser le programme pour afficher en cascade plusieurs registres.

Programmation

Dans l'exemple suivant, un bit est décalé vers la gauche à chaque seconde et le bit 0 prend l'état opposé au bit 15.



Programmation réversible

```
LDN    %SBR0.15
ST     %SBR0.0
BLK    %SBR0
LD     %S6
CU
END_BLK
```

Programmation non réversible

```
LDN    %SBR0.15
ST     %SBR0.0
LD     %S6
CU     %SBR0
```

Cas particuliers

Le tableau suivant présente une liste des cas spéciaux de fonctionnement.

Cas spécial	Description
Effet d'un redémarrage à froid (%S0=1)	Règle tous les bits du mot registre sur 0.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur les bits du mot registre.

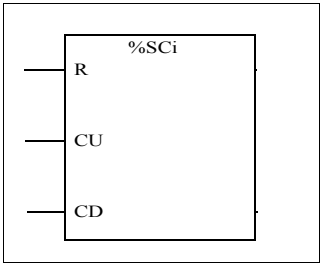
Bloc fonction pas à pas (%SCi)

Introduction

Un bloc fonction pas à pas (%SCi) permet d'accomplir une série d'étapes auxquelles des actions peuvent être affectées. Le passage d'une étape à l'autre dépend d'événements internes ou externes. Chaque fois qu'une étape est active, le bit associé (bit de fonction pas à pas %SCi.j) est réglé sur 1. La fonction pas à pas peut contrôler les bits de sortie (%Qi.j), les bits internes (%Mi) ou les bits de sortie esclave AS-Interface (%QAx.y.z). Une seule étape d'une fonction pas à pas peut être active à la fois.

Illustration

L'exemple suivant illustre un bloc fonction pas à pas.



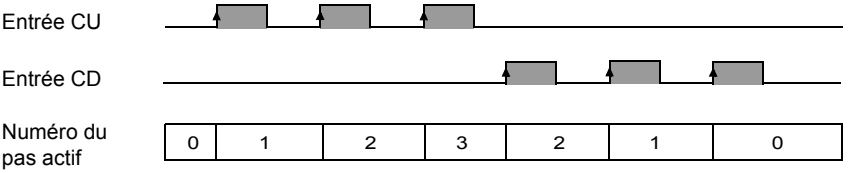
Paramètres

Le bloc fonction pas à pas possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro de fonction pas à pas	%SCi	0 - 7
Bit de fonction pas à pas	%SCi.j	Les bits de fonction pas à pas 0 à 255 (j = 0 à 255) peuvent être testés par une instruction logique de chargement et écrits à l'aide d'une instruction d'affectation.
Entrée (ou instruction) de présélection	R	Lorsque le paramètre fonction R est 1, ceci réinitialise la fonction pas à pas.
Entrée (ou instruction) d'incrémentation	CU	Sur un front montant, incrémente la fonction pas à pas d'une étape.
Entrée (ou instruction) de décrémentation	CD	Sur un front montant, décrémente la fonction pas à pas d'une étape.

Chronogramme

Le chronogramme suivant illustre le fonctionnement du bloc fonction pas à pas.

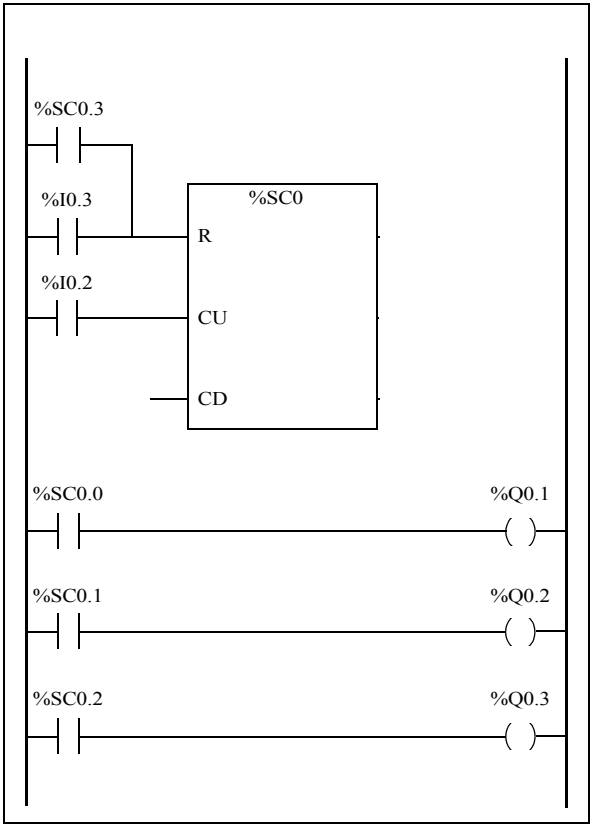


Programmation

L'exemple suivant illustre un bloc fonction pas à pas.

- La fonction pas à pas 0 est incrémentée par l'entrée %I0.2.
- La fonction pas à pas 0 est remise à 0 par l'entrée %I0.3 ou lorsqu'elle arrive à l'étape 3.
- L'étape 0 commande la sortie %Q0.1, l'étape 1 commande la sortie %Q0.2 et l'étape 2 commande la sortie %Q0.3.

L'illustration suivante présente la programmation réversible et non réversible correspondant à cet exemple.



Programmation réversible

```
BLK     %SC0
LD      %SC0.3
OR      %I0.3
R
LD      %I0.2
CU
END_BLK
LD      %SC0.0
ST      %Q0.1
LD      %SC0.1
ST      %Q0.2
LD      %SC0.2
ST      %Q0.3
```

Programmation non réversible

```
LD      %SC0.3
OR      %I0.3
R       %SC0
LD      %I0.2
CU      %SC0
LD      %SC0.0
ST      %Q0.1
LD      %SC0.1
ST      %Q0.2
LD      %SC0.2
ST      %Q0.3
```

Cas spécifique

Le tableau suivant présente une liste des cas spécifiques de fonctionnement du bloc fonction pas à pas.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Initialise la fonction pas à pas.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur la fonction pas à pas.

17.3 Traitement numérique

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre offre une introduction au traitement numérique, qui s'appuie sur des descriptions et des directives de programmation.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction aux instructions numériques	475
Instructions d'affectation	476
Instructions de comparaison	481
Instructions arithmétiques sur entiers	483
Instructions logiques	487
Instructions de décalage	488
Instructions de conversion	490
Instructions de conversion entre mots simples et doubles	492

Introduction aux instructions numériques

Présentation

Les instructions numériques s'appliquent généralement aux mots de 16 bits (voir *p. 29*) et aux doubles mots de 32 bits (voir *p. 33*). Ces instructions apparaissent entre crochets. Si le résultat de l'opération logique précédente est Vraie (accumulateur booléen = 1), l'instruction numérique est exécutée. Si ce résultat est Faux (accumulateur booléen = 0), l'instruction numérique n'est pas exécutée et l'opérande reste inchangé.

Instructions d'affectation

Présentation

Les instructions d'affectation permettent de charger l'opérande Op2 dans l'opérande Op1.

Affectation

Syntaxe des instructions d'affectation

[Op1:=Op2]	<=>	Op2 -> Op1

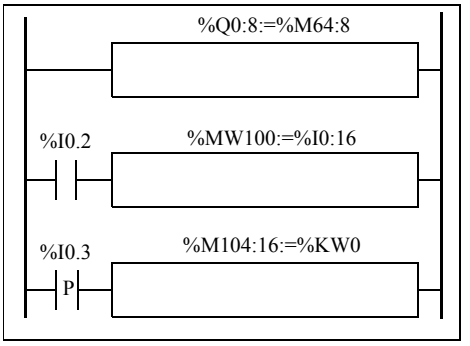
Les opérations d'affectation peuvent être exécutées sur :
- des chaînes de bits ;
- des mots ;
- des mots doubles ;
- des flottants ;
- des tables de mots ;
- des tables de mots doubles ;
- des tables de flottants.

Affectation de chaînes de bits

Les opérations peuvent être exécutées sur les chaînes de bits suivantes (voir sous-chapitre *p. 54*) :
- Chaîne de bit -> chaîne de bit (Exemple 1)
- Chaîne de bit -> mot (Exemple 2) ou mot double (indexé)
- Mot ou mot double (indexé) -> chaîne de bit (Exemple 3)
- Valeur immédiate -> chaîne de bit

Exemples

Exemples d'affectations de chaînes de bits




```
LD      1
[%Q0:8:=%M64:8]          (Ex. 1)

LD      %I0.2
[%MW100:=%I0:16]         (Ex. 2)

LDR     %I0.3
[%M104:16:=%KW0]         (Ex. 3)
```

Règles d'utilisation :

- Pour l'affectation chaîne de bit -> mot : les bits de la chaîne sont transférés vers le mot en commençant par la droite (premier bit de la chaîne vers bit 0 du mot) et les bits de mot non concernés par le transfert (longueur ≤16) sont mis à 0.
- Pour l'affectation mot -> chaîne de bits : les bits de mot sont transférés en partant de la droite (bit de mot 0 vers premier bit de la chaîne).

Affectations de chaînes de bits

Syntaxe des affectations de chaînes de bits

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
:=	[Op1 : = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).	%MWi,%QWi, %QWCi %QWAi,%SWi %MWi[%MWi], %MDi, %MDi[%MWi] %Mi:L, %Qi:L, %Si:L, %Xi:L	Valeur immédiate, %MWi, %KWi, %IW,%IWAi, %IWCi %INWi, %QWi, %QWAi %QWCi, %QNWi, %SWi, %BLK.x, %MWi[%MWi], %KWi[%MWi], %MDi[%MWi], %KDi[%MWi], %Mi:L,%Qi:L, %Si:L, %Xi:L, %Ii:L

Note : L'abréviation %BLK.x (%C0.P, par exemple) est utilisée pour décrire tout mot de bloc fonction.

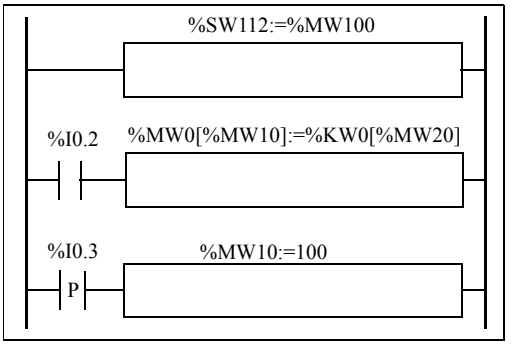
Affectation de mots

Les opérations d'affectation peuvent être exécutées sur les mots et mots doubles suivants :

- Mot (indexé) -> mot (Exemple 2) (indexé ou non)
- Mot double (indexé) -> mot double (indexé ou non)
- Valeur entière immédiate -> mot (Exemple 3) ou mot double (indexé ou non)
- Chaîne de bit -> mot ou mot double
- Flottant (indexé ou non)-> flottant (indexé ou non)
- Mot ou mot double -> chaîne de bit
- Valeur flottante immédiate -> flottant (indexé ou non)

Exemples

Exemples d'affectations de mots




```
LD      1
[%SW112:=%MW100]      (Ex. 1)

LD      %I0.2
[%MW0[%MW10]:=        (Ex. 2)
%KW0[%MW20]]

LDR     %I0.3         (Ex. 3)
[%MW10:=100]
```

Syntaxe

Syntaxe des affectations de mots

Opérateur	Format
:=	[Op1 : = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).

Le tableau suivant détaille les opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mot, mot double, chaîne de bits	%BLK.x, %MWi, %QWi, %QWAi, %QWCi, %SWi %MWi[MWi], %MDi, %MDi[%MWj], %Mi:L, %Qi:L, %Si:L, %Xi:L	Valeur immédiate, %MWi, %KWi, %IW, %IWAi, %IWCi, %QWi, %QWAi,%QWCi, %SWi, %MWi[MWi], %KWi[MWi], %MDi, %MDi[%MWj], %KDi, %KDi[MWj] %INW, %Mi:L, %Qi:L, %QNW, %Si:L, %Xi:L, %Ii:L
Flottant	%MFi, %MFi[%MWj]	Valeur flottante immédiate, %MFi, %MFi[%MWj], %KFi, %KFi[%MWj]

Note : L'abréviation %BLK.x (%R3.I, par exemple) est utilisée pour décrire tout mot de bloc fonction. Pour les chaînes de bits %Mi:L, %Si:L et %Xi:L, le repère de base du premier bit de la chaîne doit être un multiple de 8 (0, 8, 16, ..., 96, ...).

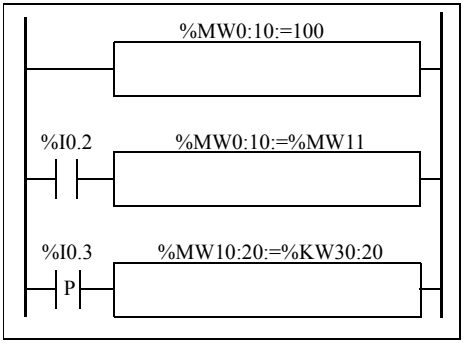
Affectation de tables de mots, de mots doubles ou de flottants

Les opérations d'affectation peuvent être exécutées sur les tables d'objets suivantes (voir *p. 55*) :

- Valeur entière immédiate -> table de mots (Exemple 1) ou de mots doubles
- Mot -> table de mots (Exemple 2)
- Table de mots -> table de mots (Exemple 3)
  La longueur de la table (L) doit être la même pour les deux tables.
- Mot double -> table de mots doubles
- Table de mots doubles -> table de mots doubles
  La longueur de la table (L) doit être la même pour les deux tables.
- Valeur flottante immédiate -> table de flottants
- Flottant -> table de flottants
- Table de flottants -> table de flottants
  La longueur de la table (L) doit être la même pour les deux tables.

Exemples

Exemples d'affectations de tables de mots




```
LD      1
[%MW0:10:=100]                  (Ex. 1)

LD      %I0.2
[%MW0:10:=%MW11]                (Ex. 2)

LDR     %I0.3
[%MW10:20:=%KW30:20]            (Ex. 3)
```

Syntaxe

Syntaxe des affectations de tables de mots, de mots doubles et de flottants

Opérateur	Format
:=	[Op1 : = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).

Le tableau suivant détaille les opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Table de mots	%MWi:L, %SWi:L	%MWi:L, %SWi:L, valeur entière immédiate, %MWi, %KWi, %IW, %QW, %IWA, %QWA, %SWi, %BLK.x
Tables de mots doubles	%MDi:L	Valeur entière immédiate, %MDi, %KDi,%MDi:L, %KDi:L
Tables de flottants	%MFi:L	Valeur flottante immédiate, %MFi, %KFi, %MFi:L, %KFi:L

Note : L'abréviation %BLK.x (%R3.I, par exemple) est utilisée pour décrire tout mot de bloc fonction.

Instructions de comparaison

Introduction

Les instructions de comparaison permettent de comparer deux opérandes.

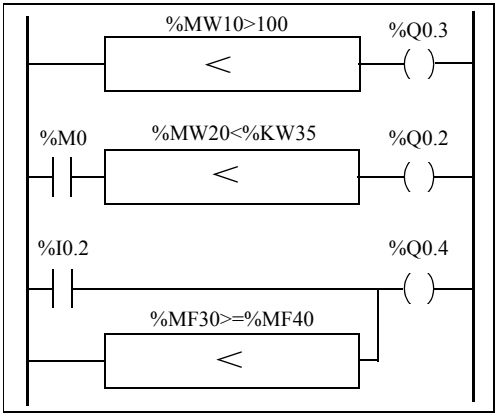
Le tableau suivant répertorie les différents types d'instructions de comparaison.

Instruction	Fonction
>	Teste si l'opérande 1 est supérieur à l'opérande 2.
>=	Teste si l'opérande 1 est supérieur ou égale à l'opérande 2.
<	Teste si l'opérande 1 est inférieur à l'opérande 2.
<=	Teste si l'opérande 1 est inférieur ou égal à l'opérande 2.
=	Teste si l'opérande 1 est égal à l'opérande 2.
<>	Teste si l'opérande 1 est différent de l'opérande 2.

Structure

La comparaison s'effectue entre les crochets qui suivent les instructions LD, AND et OR. Le résultat est 1 lorsque la comparaison requise est Vrai.

Exemples d'instructions de comparaison




```
LD    [%MW10 > 100]
ST    %Q0.3

LD    %M0
AND   [%MW20 < %KW35]
ST    %Q0.2

LD    %I0.2
OR    [%MF30>=%MF40]
ST    %Q0.4
```

Syntaxe

Syntaxe des instructions de comparaison :

Opérateur	Syntaxe
>, >=, <, <=, =, <>	LD [Op1 Opérateur Op2] AND [Op1 Opérateur Op2] OR [Op1 Opérateur Op2]

Opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi, %KWi, %INWi, %IW, %IWAi, %IWCi, %QNWi, %QWi, %QWAi, %QWCi, %QNWi, %SWi, %BLK.x	Valeur immédiate, %MWi, %KWi, %INWi, %IW, %IWAi, %IWCi, %QNWi, %QW, %QWAi, %QWCi, %SWi, %BLK.x, %MWi [%MWi], %KWi [%MWi]
Mots doubles	%MDi, %KDi	Valeur immédiate, %MDi, %KDi, %MDi [%MWi], %KD [%MWi]
Mots flottants	%MFi, %KFi	Valeur flottante immédiate, %MFi, %KFi, %MFi [%MWi], %KFi [%MWi]

Note : Les instructions de comparaison peuvent apparaître entre parenthèses.

Exemple d'utilisation d'une instruction de comparaison entre parenthèses

```
LD      %M0
AND(    [%MF20 > 10.0]
OR      %I0.0
)
ST      %Q0.1
```

Instructions arithmétiques sur entiers

Introduction

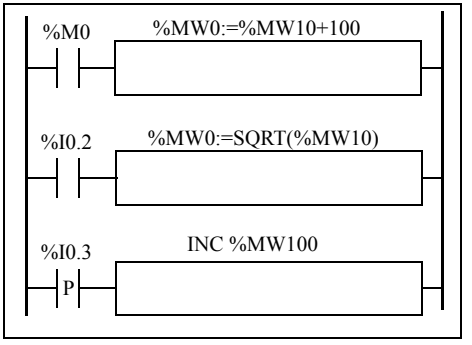
Les instructions arithmétiques permettent d'effectuer des opérations arithmétiques entre deux opérandes entiers ou sur un opérande entier.

Le tableau suivant répertorie les différents types d'instructions arithmétiques.

Instruction	Fonction
+	Addition de deux opérandes
-	Soustraction de deux opérandes
*	Multiplication de deux opérandes
/	Division de deux opérandes
REM	Reste de la division de deux opérandes
SQRT	Racine carrée d'un opérande
INC	Incrémentation d'un opérande
DEC	Décrémentation d'un opérande
ABS	Valeur absolue d'un opérande

Structure

Les opérations arithmétiques sont effectuées de la façon suivante :




```
LD      %M0
[%MW0:=%MW10 + 100]

LD      %I0.2
[%MW0:=SQRT(%MW10)]

LDR     %I0.3
[INC %MW100]
```

Syntaxe

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	**Syntaxe**
+, -, *, /, REM	[Op1: = Op 2 Opérateur Op3]
INC, DEC	[Opérateur Op1]
SQRT (1)	[Op1: = SQRT(Op2)]
ABS (1)	[Op1: = ABS(Op2)]

Opérandes :

Type	**Opérande 1 (Op1)**	**Opérandes 2 et 3 (Op2 & 3) (1)**
Mots	%MWi, %QWi, %QWAi, %QWCi, %SWi	Valeur immédiate, %MWi, %KWi, %INW, %IW, %IWAi, %IWCi, %QNW, %QW, %QWAi, %QWCi, %SWi, %BLK.x
Mots doubles	%MDi	Valeur immédiate, %MDi, %KDi

Note : (1) Avec cet opérateur, Op2 ne peut pas être une valeur immédiate.
La fonction ABS n'est utilisable qu'avec des mots doubles (%MD et %KD) et des flottants (%MF et %KF). Par conséquent OP1 et OP2 doivent être des mots doubles ou des flottants.

Débordement et conditions d'erreurs

Addition

- Débordement pendant l'opération sur mots
  Si le résultat dépasse les limites de -32 768 ou de +32 767, le bit %S18 (débordement) est mis à 1. Le résultat est alors non significatif (voir Exemple 1 page suivante). Le programme utilisateur gère le bit %S18.

Remarque :

Pour les mots doubles, les limites sont -2 147 483 648 et 2 147 483 647.

Multiplication

- Débordement pendant l'opération
  Si le résultat dépasse la capacité du mot de résultat, le bit %S18 (débordement) est mis à 1 et le résultat n'est pas significatif.

Division / reste

- Division par 0
  Si le dividende est 0, la division est impossible et le bit système %S18 est réglé sur 1. Le résultat est alors incorrect.
- Débordement pendant l'opération
  Si le quotient de la division dépasse la capacité du mot de résultat, le bit système %S18 est mis à 1.

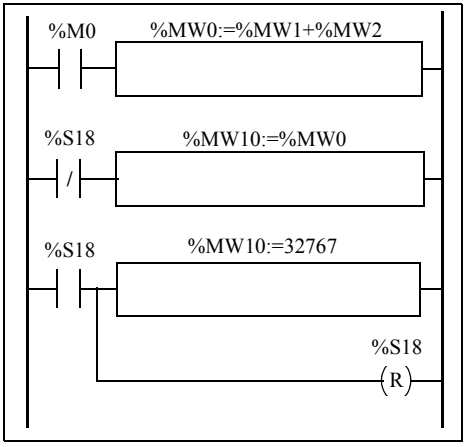
Calcul de la racine carrée

- Débordement pendant l'opération
  Le calcul de la racine carrée est uniquement effectué sur les valeurs positives. Le résultat est, par conséquent, toujours positif. Si l'opérande de racine carrée est négatif, le bit système %S18 est mis à 1 et le résultat est incorrect.

Note : Le programme utilisateur gère les bits système %S17 et %S18. L'automate les règle sur 1. Ils doivent être remis à 0 par le programme afin de pouvoir être réutilisés (voir exemple page précédente).

Exemples

Exemple 1 : débordement lors de l'addition




```
LD      %M0
[%MW0:=%MW1 + %MW2]

LDN     %S18
[%MW10:=%MW0]

LD      %S18
[%MW10:=32767]
R       %S18
```

Si %MW1 =23 241 et %MW2=21 853, le résultat réel (45 094) ne peut pas être exprimé par un mot de 16 bits, le bit %S18 est réglé sur 1 et le résultat obtenu (-20 442) est incorrect. Dans cet exemple, la valeur est fixée à 32 767 lorsque le résultat est supérieur à cette valeur.

Instructions logiques

Introduction

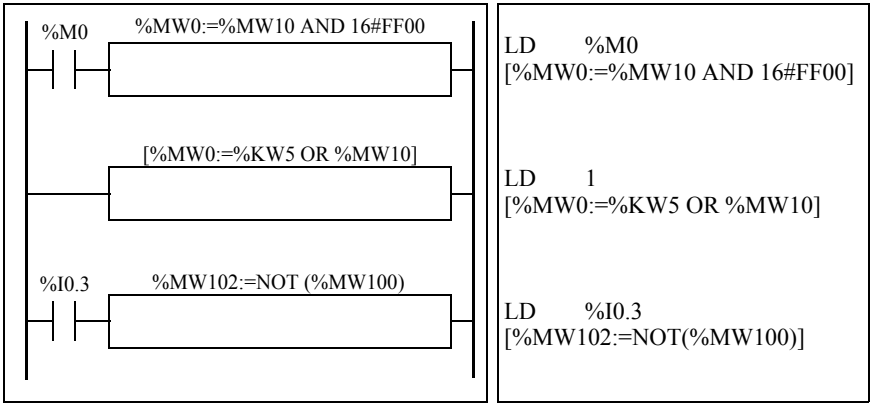
Les instructions logiques permettent d'effectuer des opérations logiques entre deux opérandes ou sur un opérande.

Le tableau suivant répertorie les différents types d'instructions logiques :

Instruction	Fonction
AND	AND (bit à bit) entre deux opérandes
OR	OR logique (bit à bit) entre deux opérandes
XOR	OR exclusif (bit à bit) entre deux opérandes
NOT	Complément logique (bit à bit) d'un opérande

Structure

Les opérations logiques sont effectuées de la façon suivante :



Syntaxe

La syntaxe dépend des opérateurs utilisés :

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérandes 2 et 3 (Op2 et 3)
AND, OR, XOR	[Op1 : = Op2 Opérateur Op3]	%MWi, %QWi, %QWAi, %QWCi, %SWi	Valeur immédiate (1), %MWi, %KWi, %IW, %IWAi, %IWCi, %QW, %QWAi, %QWCi, %SWi, %BLK.x
NOT	[Op1:=NOT(Op2)]		

Note : (1) Avec NOT, Op2 ne peut pas être une valeur immédiate.

Exemple

L'exemple suivant présente une instruction AND logique.

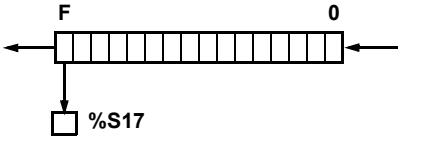
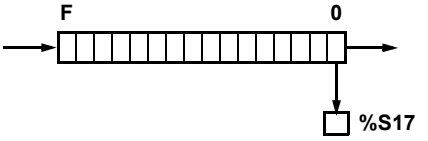
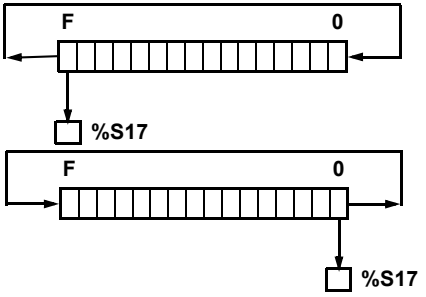
```
[%MW15:=%MW32 AND %MW12]
```

Instructions de décalage

Introduction

Les instructions de décalage déplacent les bits d'un opérande d'un certain nombre de positions vers la droite ou vers la gauche.

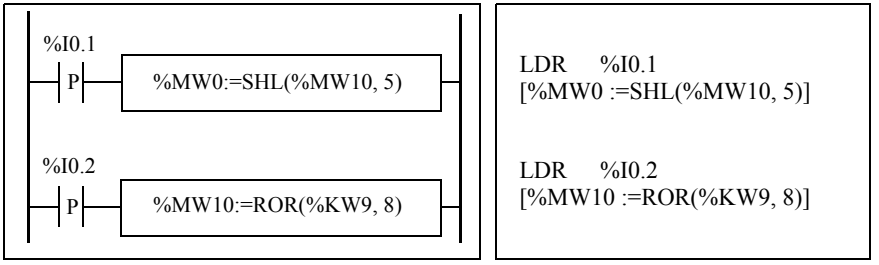
Le tableau suivant répertorie les différents types d'instructions de décalage.

Instruction	Fonction	
Décalage logique		
SHL(op2,i)	Décalage logique de i positions vers la gauche	
SHR(op2,i)	Décalage logique de i positions vers la droite	
Décalage circulaire		
ROL(op2,i)	Décalage circulaire de i positions vers la gauche	
ROL(op2,i)	Décalage circulaire de i positions vers la droite	

Note : Le bit système %S17 (voir *p. 798*) sert à indiquer le dernier bit éjecté.

Structure

Les opérations de décalage sont effectuées de la façon suivante :



Format

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	**Syntaxe**
SHL, SHR	[Op1 : = Opérateur (Op2,i)]
ROL, ROR	

Opérandes :

Types	**Opérande 1 (Op1)**	**Opérande 2 (Op2)**
Mots	%MWi, %QWi, %QWAi, %QWCi, %SWi	%MWi, %KWi, %IW, %IWAi, %IWCi, %QW, %QWAi, %QWCi, %SWi, %BLK.x
Mots doubles	%MDi	%MDi, %KDi

Instructions de conversion

Introduction

Les instructions de conversion permettent d'effectuer la conversion entre les différentes représentations numériques.

Le tableau suivant répertorie les différents types d'instructions de conversion.

Instruction	Fonction
BTI	BCD --> Conversion binaire
ITB	Binaire --> Conversion BCD

Révision du code BCD

Le codage BCD (Binary Coded Decimal - décimal codé binaire) représente les décimaux (entre 0 et 9) par un code à quatre bits. Un objet mot de 16 bits peut ainsi contenir un nombre exprimé par quatre chiffres (0000 - 9999), et un objet mot double de 32 bits peut ainsi contenir un nombre exprimé par huit chiffres.

Lors d'une conversion, le bit système %S18 est mis à 1 si la valeur n'est pas BCD. Ce bit doit être testé et remis à 0 par le programme.

Représentation BCD des décimaux :

Décimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Exemples :

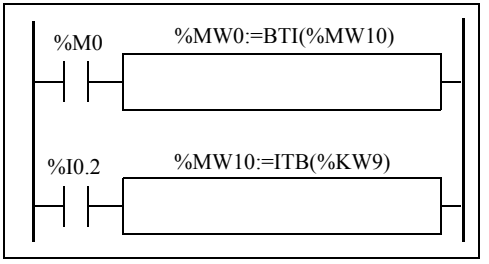
- Le mot %MW5 exprime la valeur BCD "2450", qui correspond à la valeur binaire : 0010 0100 0101 0000
- Le mot %MW12 exprime la valeur décimale "2450", qui correspond à la valeur binaire : 0000 1001 1001 0010

Le mot %MW5 est converti en mot %MW12 à l'aide de l'instruction BTI.

Le mot %MW12 est converti en mot %MW5 à l'aide de l'instruction ITB.

Structure

Les opérations de conversion sont effectuées de la façon suivante :




```
LD      %M0
[%MW0 :=BTI(%MW10)]

LD    %I0.2
[%MW10 :=ITB(%KW9)]
```

Syntaxe

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	Syntaxe
BTI, ITB	[Op1 : = Opérateur (Op2)]

Opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi, %QWi, %QWAi, %QWCi, %SWi	%MWi, %KWi, %IW, %IWAi, %IWCi, %QW, %QWAi, %QWCi, %SWi, %BLK.x
Mots doubles	%MDi	%MDi, %KDi

Exemples d'application :

L'instruction BTI peut être utilisée pour traiter une valeur de consigne aux entrées de l'automate via des roues codeuses en BCD.

L'instruction peut être utilisée pour afficher des valeurs numériques sur des afficheurs codés en BCD (résultat d'un calcul, valeur courante d'un bloc fonction, par exemple).

Instructions de conversion entre mots simples et doubles

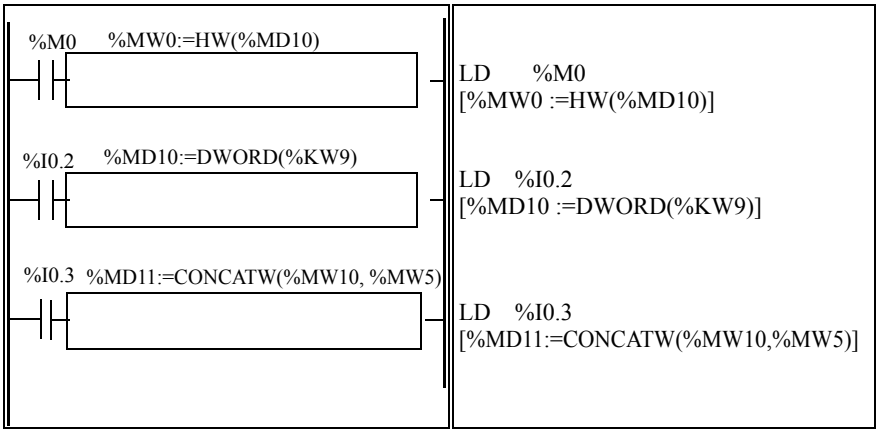
Présentation

Le tableau suivant décrit les instructions de conversion entre les mots simples et doubles :

Instruction	Fonction
LW	Extrait l'octet de poids faible d'un mot double vers un mot.
HW	Extrait l'octet de poids fort d'un mot double vers un mot.
CONCATW	Concatène deux mots pour constituer un mot double.
DWORD	Convertit un mot de 16 bits en un mot double de 32 bits.

Structure

Les opérations de conversion sont effectuées de la façon suivante :



Syntaxe

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau suivant : l

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)	Opérande 3 (Op3)
LW, HW	Op1 = Opérateur (Op2)	%MWi	%MDi, %KDi	[-]
CONCATW	Op1 = Opérateur (Op2, Op3))	%MDi	%MWi, %KWi, valeur immédiate	%MWi, %KWi, valeur immédiate
DWORD	Op1 = Opérateur (Op2)	%MDi	%MWi, %KWi	[-]

17.4 Instructions du programme

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre présente une introduction aux instructions du programme.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions END	494
Instruction NOP	496
Instructions de saut	497
Instructions de sous-programme	498

Instructions END

Introduction

Les instructions END définissent la fin de l'exécution de la scrutation d'un programme.

END, ENDC et ENDCN

Il existe trois instructions END différentes :

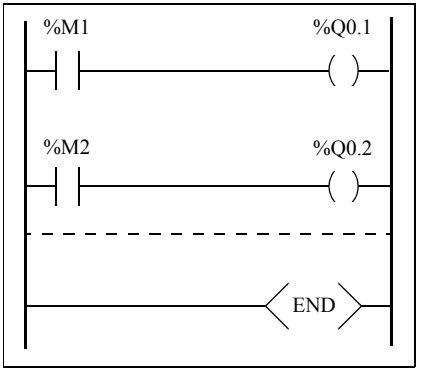
- END : fin de programme inconditionnelle
- ENDC : fin de programme si le résultat booléen de l'instruction sur test précédente est 1
- ENDCN : fin de programme si le résultat booléen de l'instruction sur test précédente est 0

Par défaut (en mode Normal), des sorties sont générées et la scrutation suivante est lancée dès la fin d'un programme.

Si la scrutation est périodique, des sorties sont générées et la scrutation suivante est lancée dès que la fin de période est atteinte.

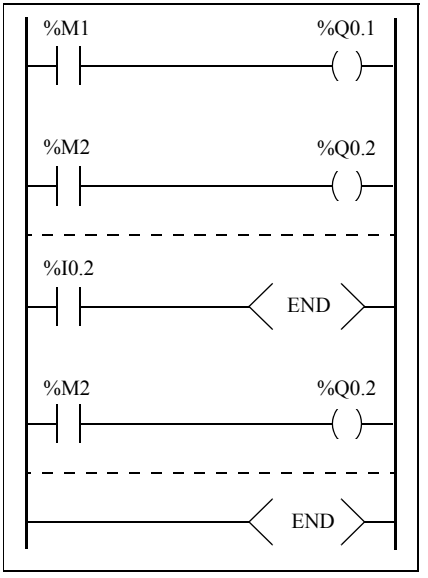
Exemples

Exemple d'instruction END inconditionnelle




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2

.....

END
```

Exemple d'instruction END conditionnelle




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2

.....

LD      %I0.2
ENDC                  → Si %I0.2 = 1, fin de scrutation du programme
LD      %M2
ST      %Q0.2

                      Si %I0.2 = 0, continue la scrutation du programme jusqu'à la nouvelle instruction END
.....

END
```

Instruction NOP

NOP

L'instruction NOP n'exécute aucune action. Utilisez cette instruction pour "réserver" des lignes d'un programme afin de pouvoir insérer ultérieurement des instructions, sans modifier les numéros de ligne.

Instructions de saut

Introduction

Les instructions de saut ont pour effet d'interrompre immédiatement l'exécution d'un programme et de le reprendre à partir de la ligne suivant la ligne contenant l'étiquette %Li (i = 1 à 16 pour un compact et de 1 à 63 pour les autres).

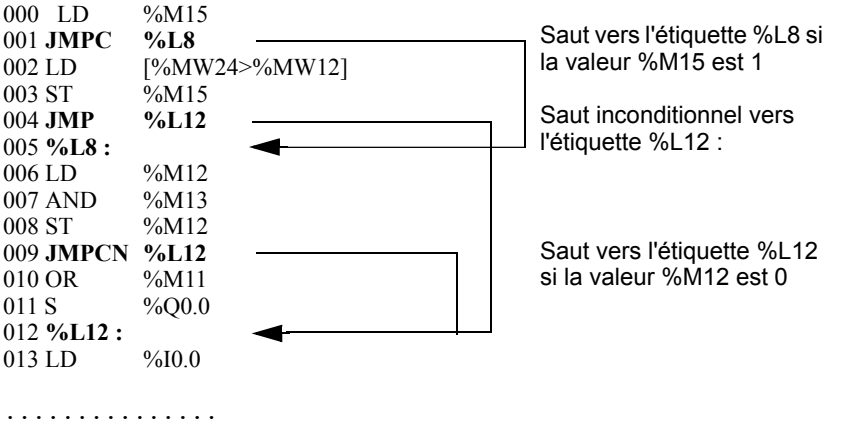
JMP, JMPC et JMPCN

Trois instructions de saut différentes sont disponibles :

- JMP : saut de programme inconditionnel
- JMPC : saut de programme si le résultat booléen de la logique précédente est 1.
- JMPCN : saut de programme si le résultat booléen de la logique précédente est 0.

Exemples

Exemples d'instructions de saut.


000 LD %M15
001 JMPC %L8
002 LD [%MW24>%MW12]
003 ST %M15
004 JMP %L12
005 %L8 :
006 LD %M12
007 AND %M13
008 ST %M12
009 JMPCN %L12
010 OR %M11
011 S %Q0.0
012 %L12 :
013 LD %I0.0
. . . . . .
Saut vers l'étiquette %L8 si la valeur %M15 est 1
Saut inconditionnel vers l'étiquette %L12 :
Saut vers l'étiquette %L12 si la valeur %M12 est 0

Directives

- Les instructions de saut sont interdites entre parenthèses et ne doivent pas être placées entre les instructions AND(, OR(et une parenthèse fermante ")".
- L'étiquette peut uniquement être placée devant une instruction LD, LDN, LDR, LDF ou BLK.
- Le numéro de l'étiquette %Li doit être défini une seule fois dans un programme.
- Le saut de programme est effectué vers une ligne de programmation en amont ou en aval. Lorsque le saut est en amont, le temps de scrutation doit être contrôlé. Un temps de scrutation trop long peut provoquer le déclenchement du chien de garde.

Instructions de sous-programme

Introduction

Les instructions de sous-programme déclenchent l'exécution d'un sous-programme, puis le retour vers le programme principal.

SRn, SRn: et RET

Les sous-programmes se composent de trois étapes :

- L'instruction **SRn** appelle le sous-programme référencé par l'étiquette SRn, si le résultat de l'instruction booléenne précédente est 1.
- Le sous-programme est référencé par l'étiquette **SRn:**, n pouvant prendre une valeur comprise entre 0 et 15 pour TWDLCAA10DRF, TWDLCAA16DRF et entre 0 et 63 pour tous les autres automates.
- L'instruction **RET** placée à la fin du sous-programme provoque le retour au programme principal.

Exemple

Exemples d'instructions de sous-programme

```
000  LD      %M15
001  AND     %M5
002  ST      %Q0.0
003  LD      [%MW24>%MW12]
004  SR8
005  LD      %I0.4
006  AND     M13
007  _
008  _
009  _
010  END

011  SR8:
012  LD      1
013  IN      %TM0
014  LD      %TM0.Q
015  ST      %M15
010  RET
```

Saut vers le sous-programme SR8

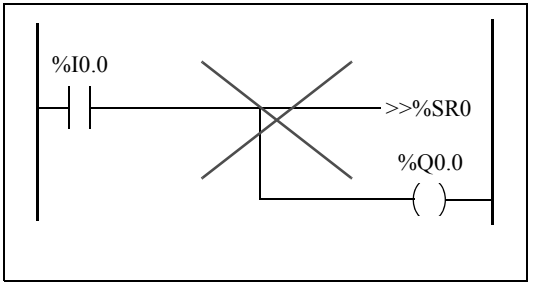
Retour au programme principal

. . . . . .

Directives

- Un sous-programme ne doit pas appeler un autre sous-programme.
- Les instructions de sous-programme sont interdites entre parenthèses et ne doivent pas être placées entre les instructions AND(, OR(et une fermeture de parenthèse ")".
- L'étiquette peut uniquement être placée devant une instruction LD ou BLK pour marquer le début d'une équation booléenne (ou d'un réseau booléen).
- L'appel du sous-programme ne doit pas être suivi d'une instruction d'affectation. En effet, le sous-programme risque de modifier le contenu de l'accumulateur booléen. Aussi celui risque d'avoir une valeur de retour différente de celle qu'il avait avant l'appel. Voir l'exemple suivant.

Exemple de programmation d'un sous-programme




```
LD      %I0.0
SR0
ST      %Q0.0

LD      %I0.0
ST      %Q0.0
SR0
```

Instructions avancées

18

Présentation

Objet de ce chapitre

Ce chapitre offre des informations sur les instructions et les blocs fonctions avancées utilisés pour créer des programmes destinés aux automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
18.1	Blocs fonctions avancés	503
18.2	Fonctions horodateur	545
18.3	Guide de démarrage rapide de l'automate PID Twido	554
18.4	Fonction PID	577
18.5	Instructions sur flottants	634
18.6	Instructions ASCII	645
18.7	Instructions sur tableaux d'objets	656

18.1 Blocs fonctions avancés

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre offre une présentation des blocs fonctions avancés et contient des exemples de programmation.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Objets mots et objets bits associés à des blocs fonction avancés	504
Principes de programmation de blocs fonctions avancés	506
Bloc fonction registre LIFO/FIFO (%Ri)	509
LIFO, fonctionnement	510
Fonctionnement FIFO	511
Programmation et configuration des registres	512
Bloc fonction %PWM (modulation de la largeur d'impulsion)	514
Bloc fonction sortie du générateur d'impulsions (%PLS)	517
Bloc fonction programmateur cyclique (%DR)	520
Fonctionnement du bloc fonction programmateur cyclique %DRi	521
Programmation et configuration des programmateurs cycliques	523
Bloc fonction compteur rapide (%FC)	525
Bloc fonction compteur rapide (%VFC)	528
Emission/réception de messages – Instruction d'échange (EXCH)	540
Bloc fonction de contrôle d'échange (%MSGx)	541

Objets mots et objets bits associés à des blocs fonction avancés

Introduction

Les blocs fonction avancés utilisent des mots et des bits dédiés de même type que les blocs fonction standards. Les blocs fonction avancés comprennent :

- les registres LIFO/FIFO (%R) ;
- les programmateurs cycliques (%DR) ;
- les compteurs rapides (%FC) ;
- les compteurs rapides (%VFC) ;
- la sortie de modulation de la largeur de l'impulsion (%PWM) ;
- la sortie du générateur d'impulsions (%PLS) ;
- le registre bits à décalage (%SBR) ;
- la fonction pas à pas (%SC) ;
- le bloc contrôle message (%MSG).

Objets accessibles par le programme

Le tableau suivant présente les mots et les bits accessibles par le programme associés aux différents blocs fonction avancés. Veuillez noter que l'accès en écriture mentionné dans le tableau suivant dépend du paramètre " Réglable ", sélectionné au moment de la configuration. Ce réglage permet d'autoriser ou de refuser l'accès aux mots ou aux bits par TwidoSuite ou par l'interface opérateur.

Bloc fonction avancé	Mots et bits associés		Repère	Accès en mode écriture
%R	Mot	Entrée du registre	%Ri.I	Oui
	Mot	Sortie du registre	%Ri.O	Oui
	Bit	Sortie registre plein	%Ri.F	Non
	Bit	Sortie registre vide	%Ri.E	Non
%DR	Mot	Numéro du pas courant	%DRi.S	Oui
	Bit	Dernier pas égal au pas courant	%DRi.F	Non
%FC	Mot	Valeur courante	%FCi.V	Oui
	Mot	Valeur de présélection	%FCi.P	Oui
	Bit	Terminé	%FCi.D	Non

Bloc fonction avancé	Mots et bits associés		Repère	Accès en mode écriture
%VFC	Mot	Valeur courante	%VFCi.V	Non
	Mot	Valeur de présélection	%VFCi.P	Oui
	Bit	Sens de comptage	%VFCi.U	Non
	Mot	Valeur de capture	%VFCi.C	Non
	Mot	Valeur de seuil 0	%VFCi.S0	Oui
	Mot	Valeur de seuil 1	%VFCi.S1	Oui
	Bit	Sortie pour	%VFCi.F	Non
	Bit	Sortie réflexe 0 activée	%VFCi.R	Oui
	Bit	Sortie réflexe 1 activée	%VFCi.S	Oui
	Bit	Sortie seuil 0	%VFCi.TH0	Non
	Bit	Sortie seuil 1	%VFCi.TH1	Non
	Bit	Base temps de la mesure de fréquence	%VFCi.T	Oui
%PWM	Mot	Pourcentage d'impulsions au pas 1 par rapport à la période totale.	%PWMi.R	Oui
	Mot	Période préréglée	%PWMi.P	Oui
%PLS	Mot	Nombre d'impulsions	%PLSi.N	Oui
	Mot	Valeur de présélection	%PLSi.P	Oui
	Bit	Sortie courante activée	%PLSi.Q	Non
	Bit	Génération terminée	%PLSi.D	Non
%SBR	Bit	Bit de registre	%SBRi.J	Non
%SC	Bit	Bit de compteur à pas	%SCi.j	Oui
%MSG	Bit	Terminé	%MSGi.D	Non
	Bit	Erreur	%MSGi.E	Non

Principes de programmation de blocs fonctions avancés

Présentation

Les applications Twido sont stockées sous la forme de programmes en langage liste d'instructions, et ce, même si ces applications ont été rédigées à l'aide d'un Editeur Ladder. Les automates Twido peuvent ainsi être considérées comme des "machines à listes". Le terme "réversibilité" se rapporte à la possibilité offerte par TwidoSuite de représenter une application sous forme de programme en langage liste d'instructions en une application écrite en langage schéma à contacts, puis de revenir en arrière. Par défaut, tous les programmes schémas à contacts sont réversibles.

Tout comme les blocs fonctions élémentaires, les blocs fonctions avancés doivent se conformer à des règles de réversibilité. La structure des blocs fonctions réversibles dans le langage liste d'instructions requiert l'utilisation des instructions suivantes :

- **BLK** : marque le début du bloc et la section d'entrée du bloc fonction.
- **OUT_BLK** : marque le début de la section de sortie du bloc fonction.
- **END_BLK** : marque la fin du bloc fonction.

Note : Il n'est pas nécessaire d'utiliser ces instructions de blocs fonctions réversibles pour un programme par listes d'instructions qui fonctionne correctement. Certaines instructions permettent une programmation en langage liste d'instructions non réversible.

Entrées et sorties dédiées

Les fonctions avancées Compteur rapide (FC), Compteur très rapide (VFC), PLS et PWM utilisent des entrées et des sorties dédiées. Ces bits ne sont toutefois pas réservés à une utilisation exclusive par un bloc unique. Au contraire, l'utilisation de ces ressources dédiées doit faire l'objet d'une gestion spécifique.

Lorsque vous utilisez des fonctions avancées, il est nécessaire que vous gériez la méthode d'allocation des entrées et des sorties dédiées. TwidoSuite vous assiste lors de la configuration de ces ressources en affichant des informations de configuration d'E/S et en vous avertissant si une entrée ou une sortie dédiée est déjà utilisée par un bloc fonction configuré.

Le tableau suivant résume les dépendances des entrées et des sorties dédiées, ainsi que les fonctions spécifiques.

En cas d'utilisation avec des fonctions de comptage :

Entrées	Utilisation
%I0.0.0	%VFC0: Gestion Haut/Bas ou Phase B
%I0.0.1	%VFC0: Entrée d'impulsion ou phase A
%I0.0.2	%FC0: Entrée d'impulsion ou entrée de présélection %VFC0
%I0.0.3	%FC1: Entrée d'impulsion ou entrée de capture %VFC0
%I0.0.4	%FC2: Entrée d'impulsion ou entrée de capture %VFC1
%I0.0.5	Entrée de présélection %VFC1
%I0.0.6	%VFC1: Gestion Haut/Bas ou Phase B
%I0.0.7	%VFC1: Entrée d'impulsion ou phase A

En cas d'utilisation avec des fonctions de comptage ou des fonctions spéciales :

Sorties	Utilisation
%Q0.0.0	Sortie %PLS0 ou PWM0
%Q0.0.1	Sortie %PLS1 ou PWM1
%Q0.0.2	Sorties réflexes pour %VFC0
%Q0.0.3	
%Q0.0.4	Sorties réflexes pour %VFC1
%Q0.0.5	

Utilisation d'entrées et de sorties dédiées

TwidoSuite utilise les règles suivantes lors de l'utilisation d'entrées et de sorties dédiées.

- Chaque bloc fonction utilisant des E/S dédiées doit être configuré et référencé dans l'application. L'E/S est uniquement allouée lors de la configuration d'un bloc fonction. Elle ne l'est pas lors de son référencement dans un programme.
- Après qu'un bloc fonction a été configuré, son entrée et sa sortie dédiées ne peuvent pas être utilisées par l'application ou par un autre bloc fonction.
  Par exemple, si vous configurez %PLS0, vous ne pouvez pas utiliser %Q0.0.0 dans %DR0 (programmateur cyclique) ou dans la logique de l'application (ST %Q0.0.0).
- Si une entrée ou une sortie dédiée est requise par un bloc fonction déjà utilisé par l'application ou par un autre bloc fonction, il n'est pas possible de configurer ce bloc fonction.
  Par exemple, si vous configurez %FC0 comme compteur, %VFC0 ne pourra pas être configuré pour utiliser %I0.0.2 comme entrée de capture.

Note : Pour modifier l'utilisation des E/S dédiées, vous devez d'abord supprimer la configuration du bloc fonction en définissant le type d'objet sur "non utilisé", puis supprimer les références au bloc fonction dans votre application.

Bloc fonction registre LIFO/FIFO (%Ri)

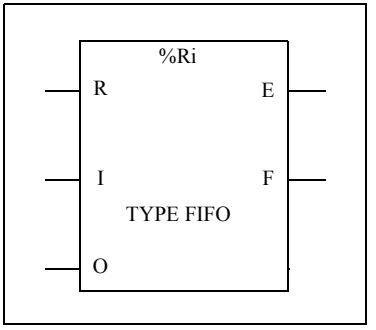
Introduction

Un registre est un bloc mémoire qui permet de stocker jusqu'à 16 mots de 16 bits de deux manières différentes :

- par une file d'attente, appelée "FIFO" (First In, First Out – Premier entré, Premier sorti) ;
- par une pile, appelée "LIFO" (Last In, First Out – Dernier entré, Premier sorti).

Illustration

L'exemple suivant illustre l'utilisation du bloc fonction registre.



Bloc fonction registre

Paramètres

Le bloc fonction registre possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro de registre	%Ri	0 à 3
Type	FIFO ou LIFO	File d'attente ou Pile)
Mot d'entrée	%Ri.I	Mot d'entrée du registre. Peut être lu, testé et écrit.
Mot de sortie	%Ri.O	Mot de sortie du registre. Peut être lu, testé et écrit.
Entrée (ou instruction) de stockage	I (In, Entrée)	Sur un front montant, stocke le contenu du mot %Ri.I dans le registre.
Entrée (ou instruction) de récupération	O (Out, Sortie)	Sur un front montant, charge un mot de données du registre dans le mot %Ri.O.
Entrée (ou instruction) RAZ	R (Remise à zéro)	A l'état 1, initialise le registre.
Sortie "Vide"	E (vide)	Le bit %Ri.E associé indique que le registre est vide. Peut être testé.
Sortie "Plein"	F (plein)	Le bit %Ri.F associé indique que le registre est plein. Peut être testé.

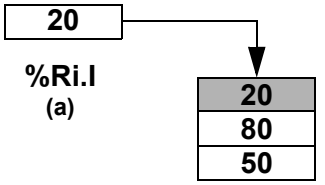
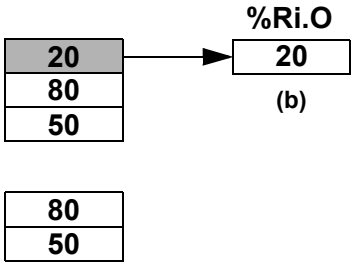
LIFO, fonctionnement

Introduction

En fonctionnement LIFO (Last In, First Out - Dernier entré, Premier sorti), la dernière information entrée est la première à être récupérée.

Fonctionnement

Le tableau suivant décrit le fonctionnement LIFO.

Etape	Description	Exemple
1	A la réception d'une demande de stockage (front montant sur l'entrée I ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I (qui a préalablement été chargé) est stocké au plus haut de la pile (fig. a). Lorsque la pile est pleine (sortie F=1), plus aucun élément ne peut être stocké.	Stockage du contenu de %Ri.I en haut de la pile. 
2	A la réception d'une demande de récupération (front montant sur l'entrée O ou activation de l'instruction O), le mot de données le plus haut (le dernier à avoir été entré) est chargé dans le mot %Ri.0 (fig. b). Lorsque le registre est vide (sortie E=1), plus aucun élément ne peut être récupéré. Le mot de sortie %Ri.O n'est pas modifié et sa valeur reste inchangée.	Récupération du mot de données au plus haut de la pile. 
3	La pile peut être réinitialisée à tout moment (état 1 sur l'entrée R ou activation de l'instruction R). L'élément indiqué par le pointeur est alors le plus haut dans la pile.	

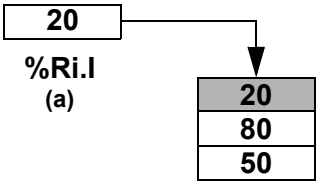
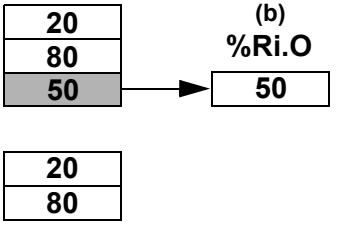
Fonctionnement FIFO

Présentation

En fonctionnement FIFO (First In, First Out - Premier entré, Premier sorti), la première information entrée est la première à être récupérée.

Fonctionnement

Le tableau suivant décrit le fonctionnement FIFO.

Etape	Description	Exemple
1	A la réception d'une demande de stockage (front montant sur l'entrée I ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I (qui a préalablement été chargé) est stocké au plus haut de la file d'attente (fig. a). Lorsque la file d'attente est pleine (sortie F=1), plus aucun élément ne peut être stocké.	Stockage du contenu de %Ri.I en haut de la file d'attente. 
2	A la réception d'une demande de récupération (front montant sur l'entrée O ou activation de l'instruction O), le mot de données le moins haut dans la file d'attente est chargé dans le mot de sortie %Ri.O et le contenu du registre est déplacé d'une place vers le bas, dans la file d'attente (fig. b). Lorsque le registre est vide (sortie E=1), plus aucun élément ne peut être récupéré. Le mot de sortie %Ri.O n'est pas modifié et sa valeur reste inchangée.	Récupération de la première information qui est ensuite chargée dans %Ri.O. 
3	La file d'attente peut être réinitialisée à tout moment (état 1 sur l'entrée R ou activation de l'instruction R).	

Programmation et configuration des registres

Introduction

L'exemple de programmation suivant illustre le chargement du contenu d'un mot mémoire (%MW34) dans un registre (%R2.I) lors d'une demande de stockage (%I0.2), si le registre %R2 n'est pas plein (%R2.F = 0). La demande de stockage dans le registre est effectuée par %M1. La demande de récupération est effectuée par l'entrée %I0.3 et %R2.O est chargé dans %MW20, si le registre n'est pas vide (%R2.E = 0).

Exemple de programmation

L'illustration suivante représente un bloc fonction registre et présente des exemples de programmation réversible et non réversible.

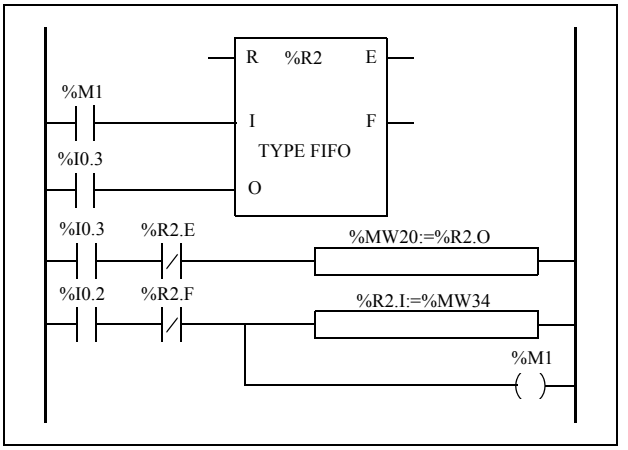


Schéma à contacts

```
BLK          %R2
LD           %M1
I
LD           %I0.3
O
END_BLK
LD           %I0.3
ANDN         %R2.E
[%MW20:=%R2.O]
LD           %I0.2
ANDN         %R2.F
[%R2.I:=%MW34]
ST           %M1
```

Programme réversible

```
LD           %M1
I            %R2
LD           %I0.3
O            %R2
ANDN         %R2.E
[%MW20:=%R2.O]
LD           %I0.2
ANDN         %R2.F
[%R2.I:=%MW34]
ST           %M1
```

Programme non réversible

Configuration

Seul le type du registre devra être entré au cours de la configuration.

- FIFO (par défaut), ou
- LIFO

Cas particuliers

Le tableau suivant présente une liste de cas spécifiques de fonctionnement :

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Provoque l'initialisation du contenu du registre. Le bit de sortie %Ri.E associé à la sortie E est mis à 1.
Effet d'une reprise à chaud (%S1=1) d'un arrêt de l'automate	N'a aucun effet sur la valeur courante du registre ou sur l'état de ses bits de sortie.

Bloc fonction %PWM (modulation de la largeur d'impulsion)

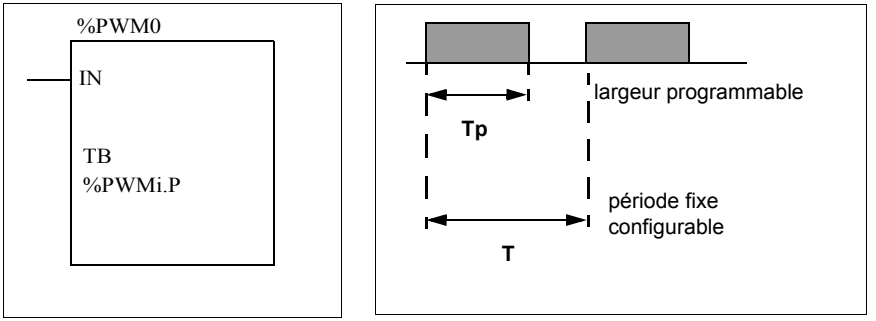
Introduction

Le bloc fonction de modulation de la largeur d'impulsion (%PWM) génère un signal d'onde carrée sur des voies de sortie dédiées (%Q0.0.0 ou %Q0.0.1) avec une largeur variable et par conséquent, un cycle d'activité. Les automates disposant de sorties relais pour ces deux voies ne prennent pas en charge cette fonction, en raison d'une limitation de fréquences.

Deux blocs %PWM sont disponibles. %PWM0 utilise la sortie dédiée %Q0.0.0 et %PMW1 utilise la sortie dédiée %Q0.0.1. Les blocs fonction %PLS se partagent les mêmes sorties dédiées. Il est donc nécessaire de choisir l'une ou l'autre des fonctions.

Illustration

Bloc PWM et chronogramme :



Paramètres

Le tableau suivant présente les différents paramètres du bloc fonction PWM.

Paramètre	Etiquette	Description
Base temps	TB	0,142 ms, 0,57 ms, 10 ms, 1 s (valeur par défaut)
Présélection de la période	%PWMi.P	0 < %PWMi.P <= 32767 avec une base temps de 10 ms ou 1 s 0 < %PWMi.P <= 255 avec une base temps de 0,57 ms ou 0,142 ms 0 = Fonction non utilisée
Rapport cyclique	%PWMi.R	Cette valeur donne le pourcentage du signal à l'état 1 au cours d'une période. Le Tp de largeur est ainsi égal à : Tp = T * (%PWMi.R/100). L'application utilisateur écrit la valeur pour %PWMi.R. Ce mot contrôle le cycle d'activité de la période. Pour plus d'informations sur la définition T, reportez-vous à la section suivante, intitulée "Plage de périodes". La valeur par défaut est 0 et les valeurs supérieures à 100 sont considérées comme étant égales à 100.
Entrée générateur d'impulsions	IN	A l'état 1, le signal de modulation de la largeur d'impulsion est généré sur la voie de sortie. A l'état 0, la voie de sortie est paramétrée sur 0.

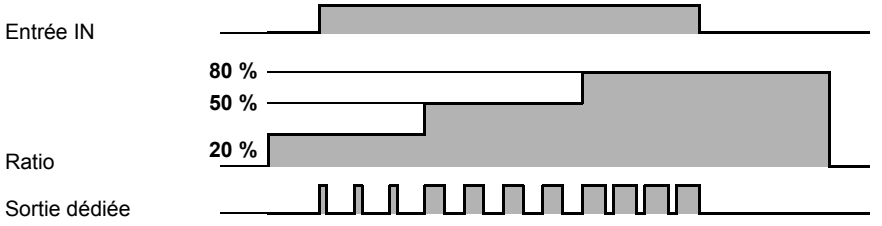
Plage de périodes

La valeur de présélection et la base temps peuvent être modifiées au moment de la configuration. Ces paramètres sont utilisés pour fixer la période du signal T=%PWMi.P * TB. L'obtention de rapports bas nécessite que le %PWMi.P sélectionné soit d'autant plus élevé. Plage de périodes disponible :

- 0,142 ms à 36,5 ms en pas de 0,142 ms (27,4 Hz à 7 kHz)
- 0,57 ms à 146 ms en pas de 0,57 ms (6,84 Hz à 1,75 kHz)
- 10 ms à 5,45 min en pas de 10 ms
- 1 s à 9,1 heures en pas de 1 s

Fonctionnement

La fréquence du signal de sortie se règle pendant la configuration en sélectionnant la base de temps TB et la présélection %PWMi.P. Le fait de modifier le cycle d'activité de PWMi.R dans le programme module la largeur du signal. L'illustration suivante représente un diagramme d'impulsion du bloc fonction PWM avec différents cycles de charge.



Programmation et configuration

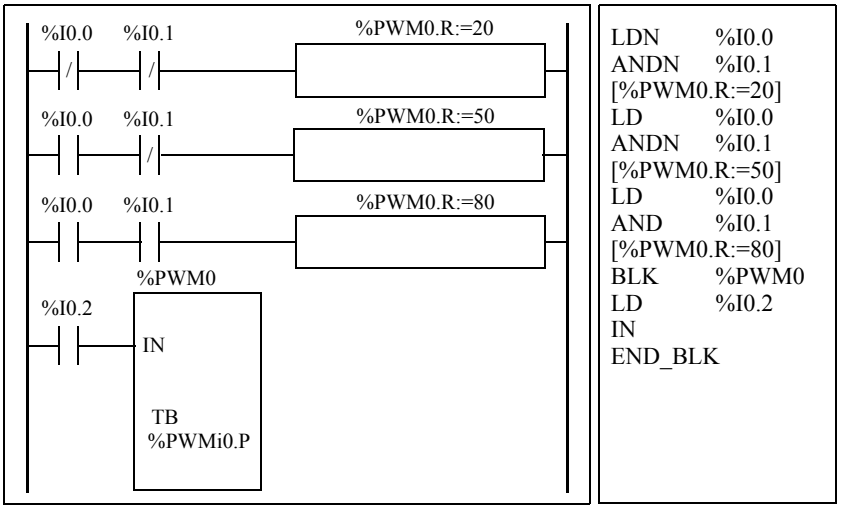
Dans cet exemple, la largeur du signal est modifiée par le programme en fonction de l'état des entrées %I0.0.0 et %I0.0.1 de l'automate.

Si %I0.0.1 et %I0.0.2 sont réglés sur 0, le rapport %PWM0.R est réglé sur 20 % et la durée du signal à l'état 1 est alors égale à : 20 % x 500 ms = 100 ms.

Si %I0.0.0 est réglé sur 0 et %I0.0.1 est réglé sur 1, le rapport %PWM0.R est réglé sur 50 % (durée de 250 ms).

Si %I0.0.0 et %I0.0.1 sont réglés sur 1, le rapport %PWM0.R est réglé sur 80 % (durée de 400 ms).

Exemple de programmation :



Cas particuliers

Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction PWM.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Règle le rapport %PWMi.R sur 0. En complément, la valeur de %PWMi.P est rétablie sur sa valeur configurée d'origine et prévaudra sur toute modification apportée dans l'éditeur de tables d'animation ou l'afficheur optionnel.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet
Incidence du fait que les sorties sont dédiées au bloc %PWM	Le fait de forcer la sortie %Q0.0.0 ou %Q0.0.1 à l'aide d'un dispositif de programmation n'interrompt pas la génération du signal.

Bloc fonction sortie du générateur d'impulsions (%PLS)

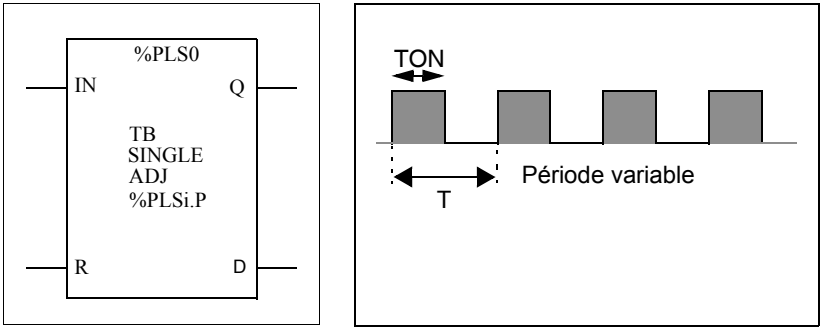
Introduction

Le bloc fonction %PLS est utilisé pour générer des signaux carrés. Il existe deux fonctions %PLS disponibles sur les voies de sortie dédiées %Q0.0.0 ou %Q0.0.1. Le bloc fonction %PLS autorise seulement une largeur de signal unique ou un cycle d'activité de 50 %. Vous pouvez choisir de limiter le nombre d'impulsions ou le moment où le train d'impulsion est exécuté. Cela peut être déterminé au moment de la configuration et/ou de la mise à jour par l'application utilisateur.

Note : Les automates disposant de sorties relais pour ces deux voies ne prennent pas en charge cette fonction %PLS.

Représentation

Exemple de bloc fonction du générateur d'impulsions en mode standard :



- TON = T/2 pour les bases temps 0,142 ms et 0,57 ms
  = (%PLSi.P*TB)/2
- TON=[partie entière (%PLSi.P)/2]*TB pour les bases temps 10 ms à 1 s.

Caractéristiques Le tableau suivant présente les caractéristiques du bloc fonction PLS :

Fonction	Objet	Description
Base temps	TB	0,142 ms, 0,57 ms, 10 ms, 1 s
Période préréglée	%PLSi.P	Les impulsions sur la sortie %PLS1 ne sont pas arrêtées lorsque %PLS1.N ou %PLS1.ND* est atteint pour les bases temps 0,142 ms et 0,57 ms. ● 1 < %PLSi.P <= 32767 pour une base temps de 10 ms ou 1 sec ● 0 < %PLSi.P <= 255 pour une base temps de 0,57 ms ou 0,142 ms ● 0 = Fonction non utilisée. Pour obtenir une bonne précision de rapport cyclique avec les bases temps de 10 ms et 1 s, il est conseillé d'avoir un %PLSi >= 100 si P est impaire.
Nombre d'impulsions	%PLSi.N %PLSi.ND *	Le nombre d'impulsions à générer sur une période T peut être limité à 0 <= %PLSi.N <= 32 767 en mode standard ou à 0 <= %PLSi.ND <= 4 294 967 295 en mode double mot. La valeur par défaut est mise à 0. Pour produire un nombre illimité d'impulsions, réglez %PLSi.N ou %PLSi.ND sur zéro. Il est toujours possible de modifier le nombre d'impulsions sans tenir compte du paramétrage de l'option Réglable.
Réglable	Y/N	Lorsque défini sur Y (Oui), il est possible de modifier la valeur de présélection %PLSi.P via l'IHM ou l'éditeur de tables d'animation. Lorsque défini sur N (Non), il est impossible d'accéder à cette présélection.
Entrée générateur d'impulsions	IN	A l'état 1, la génération des impulsions se fait sur la voie de sortie dédiée. A l'état 0, la voie de sortie est paramétrée sur 0.
Entrée RAZ	R	A l'état 1, les sorties %PLSi.Q et %PLSi.D sont paramétrées sur 0. Le nombre d'impulsions générées sur une période T est paramétré sur 0.
Génération d'impulsions sur sortie courante	%PLSi.Q	A l'état 1, le signal des impulsions est généré sur la voie de sortie dédiée configurée.
Sortie de génération d'impulsions terminée	%PLSi.D	A l'état 1, la génération du signal est terminée. Le nombre voulu d'impulsions a été généré.

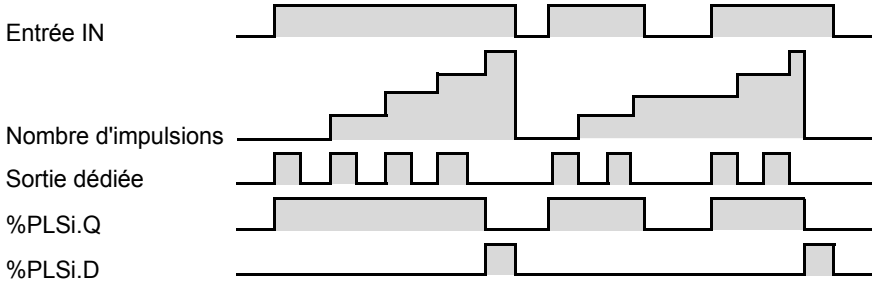
Note : (*) correspond à une variable de mot double.

Plage de périodes

La valeur de présélection et la base temps peuvent être modifiées au moment de la configuration. Ces paramètres sont utilisés pour fixer la période du signal T=%PLSi.P * BT. Plage de périodes disponible :

- 0,142 ms à 36,5 ms en pas de 0,142 ms (27,4 Hz à 7 kHz)
- 0,57 ms à 146 ms en pas de 0,57 ms (6,84 Hz à 1,75 kHz)
- 20 ms à 5,45 min en pas de 10 ms
- 2 s à 9,1 heures en pas de 1 s

Fonctionnement L'exemple suivant illustre le bloc fonction %PLS.



Cas particuliers

Cas particulier	Description
Effet d'une reprise à froid (%S0=1)	Règle la fonction %PLSi.P sur la valeur définie au cours de la configuration.
Effet d'une reprise à chaud (%S1=1)	Aucun effet
Effet de la modification de la valeur de présélection (%PLSi.P)	Prend effet immédiatement
Incidence du fait que les sorties sont dédiées au bloc %PLS	Le fait de forcer la sortie %Q0.0.0 ou %Q0.0.1 à l'aide d'un dispositif de programmation n'interrompt pas la génération du signal.

Note : %PLSx.D est défini lorsque le nombre voulu d'impulsions a été généré. Il est ensuite remis à zéro en réinitialisant les entrées IN ou R sur 1.

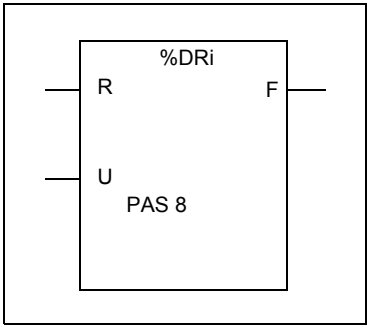
Bloc fonction programmateur cyclique (%DR)

Introduction

Le fonctionnement des programmateurs cycliques est semblable à celui des programmateurs cycliques électromécaniques qui permettent la modification de pas en fonction d'événements externes. A chaque pas, le point haut d'une came donne une commande exécutée par l'automate. Dans le cas d'un programmateur cyclique, ces points hauts sont symbolisés par l'état 1 pour chacun des pas et sont affectés aux bits de sortie %Qi.j, aux bits internes %Mi ou aux bits de sortie esclave AS-Interface %QAx.y.z, appelés "bits de contrôle".

Illustration

L'exemple suivant illustre l'utilisation du bloc fonction programmateur cyclique.



Bloc fonction programmateur cyclique

Paramètres

Le bloc fonction programmateur cyclique possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro	%DRi	0 à 3 automates compacts 0 à 7 automates modulaires
Numéro du pas courant	%DRi.S	0<%DRi.S<7. Mot pouvant être lu et écrit. La valeur écrite doit être une valeur décimale immédiate. Une fois écrite, la valeur sera prise en compte à la prochaine exécution du bloc fonction.
Nombre de pas		1 à 8 (par défaut)
Entrée retour au pas 0(ou à l'instruction)	R (Reset)	A l'état 1, règle le programmateur cyclique sur le pas 0.
Entrée (ou instruction) avancée	U (haut)	Sur un front montant, provoque le passage du programmateur cyclique au pas suivant et met à jour les bits de contrôle.
Sortie	F (plein)	Indique que le pas courant est égal au dernier pas défini. Le bit associé %DRi.F peut être testé (par exemple, %DRi.F=1, si %DRi.S= nombre de pas configurés - 1).
Bits de contrôle		Bits de sortie ou bits internes associés au pas (16 bits de contrôle) et définis dans l'éditeur de configuration.

Fonctionnement du bloc fonction programmateur cyclique %DRi

Introduction

Le programmateur cyclique comprend :

- une matrice de données constantes (des cames), organisée en huit pas (numérotés de 0 à 7) et 16 bits de données (état du pas), disposés en colonnes numérotées de 0 à F ;
- une liste des bits de contrôle est associée à une sortie configurée (%Qi.j.k), à un mot mémoire (%Mi) ou à la sortie esclave AS-Interface (%QAx.y.z). Au cours du pas courant, les bits de contrôle prennent les états binaires définis pour ce pas.

L'exemple présenté dans le tableau suivant résume les caractéristiques principales du programmateur cyclique.

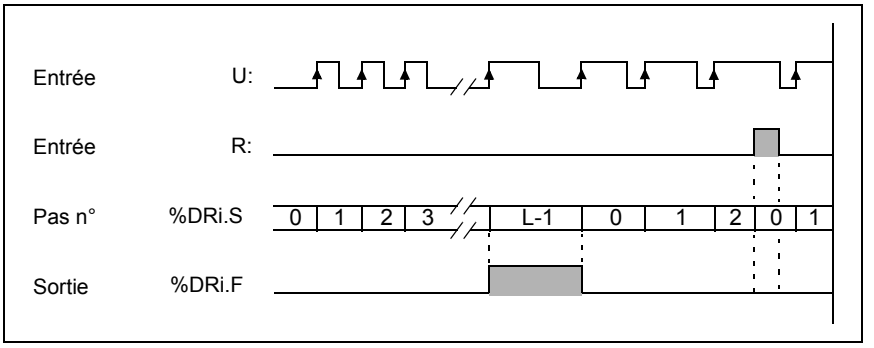
Colonne	**0**	**1**	**2**		**D**	**O**	**F**
Bits de contrôle	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Pas 0	0	0	1		1	1	0
Pas 1	1	0	1		1	0	0
Pas 5	1	1	1		0	0	0
Pas 6	0	1	1		0	1	0
Pas 7	1	1	1		1	0	0

Fonctionnement

Dans l'exemple précédent, le pas 5 est le pas courant, les bits de contrôle %Q0.1, %Q0.3 et %Q1.5 sont à l'état 1 ; les bits de contrôle %Q0.6, %Q0.5 et %Q1.0 sont à l'état 0. Le numéro du pas courant est incrémenté d'une unité sur chaque front montant à l'entrée U (ou lors de l'activation de l'instruction U). Le pas courant peut être modifié par le programme.

Chronogramme

Le chronogramme suivant illustre le fonctionnement du programmateur cyclique.



Cas particuliers

Le tableau suivant présente une liste des cas spécifiques de fonctionnement du programmateur cyclique.

Cas spécifique	Description
Effets d'un redémarrage à froid (%S0=1)	Provoque la réinitialisation du programmateur cyclique au pas 0 (mise à jour des bits de contrôle).
Effet d'une reprise à chaud (%S1=1)	Met à jour les bits de contrôle d'après le pas courant.
Effet d'un saut de programme	Le fait de ne plus scruter le programmateur cyclique ne remet pas les bits de contrôle à zéro.
Mise à jour des bits de contrôle	Survient uniquement en cas de changement de pas ou lors d'un démarrage à froid ou d'un redémarrage à chaud.

Programmation et configuration des programmateurs cycliques

Introduction

L'exemple suivant illustre la programmation et la configuration d'un programmateur cyclique. Les six premières sorties (%Q0.0 à %Q0.5) sont activées les unes à la suite des autres, chaque fois que l'entrée %I0.1 est réglée sur 1. L'entrée I0.0 remet les sorties à zéro.

Exemple de programmation

L'illustration suivante représente un bloc fonction programmateur cyclique et présente des exemples de programmation réversible et non réversible.

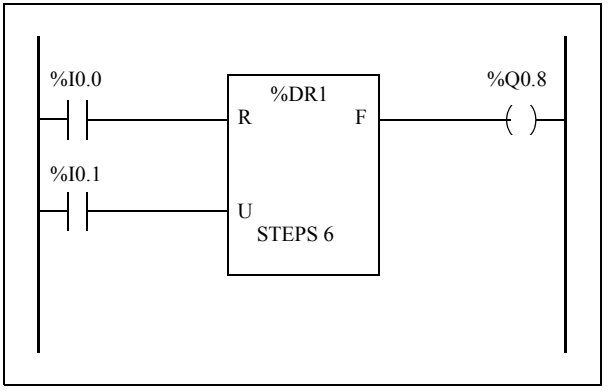


Schéma à contacts

```
BLK     %DR1
LD      %I0.0
R
LD      %I0.1
U
OUT_BLK
LD      F
ST      %Q0.8
END_BLK
```

Configuration

Les informations suivantes sont définies au moment de la configuration :

- nombre de pas : 6
- états de sortie (bits de contrôle) pour chaque pas du programmateur cyclique

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Etape 1 :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 2 :	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 3 :	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 4 :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Etape 5 :	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Etape 6 :	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

- Affectation des bits de contrôle

1:	%Q0.0	4:	%Q0.1
2:	%Q0.2	5:	%Q0.3
3:	%Q0.4	6:	%Q0.5

Bloc fonction compteur rapide (%FC)

Introduction

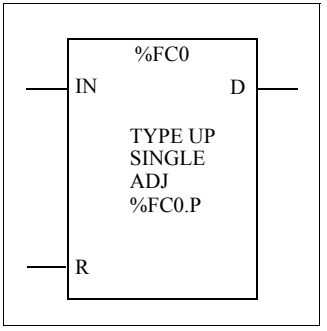
Le bloc fonction compteur rapide (%FC) sert à la fois de compteur et de décompteur. Il peut compter le front montant des entrées TOR pour des fréquences allant jusqu'à 5 kHz(1) en mode de calcul mot simple ou mot double. Etant donné que les compteurs rapides (FC) sont gérés par des interruptions matérielles spécifiques, le maintien du taux d'échantillonnage maximal des fréquences peut varier en fonction de la configuration de votre application et de votre matériel.

Note : (1) Pour le module TWDLEDCK1 Twido Extreme, le compteur rapide peut compter le front montant à des fréquences pouvant atteindre jusqu'à 10 kHz.

Les automates compacts TWDLC••40DRF peuvent contenir jusqu'à quatre compteurs rapides alors que toutes les autres gammes d'automates compacts ne peuvent être configurées que pour utiliser au maximum trois compteurs rapides. Quant aux automates modulaires, ils ne peuvent en comporter que deux. Les blocs fonction compteur rapide %FC0, %FC1, %FC2 et %FC3 utilisent respectivement les entrées dédiées %I0.0.2, %I0.0.3, %I0.0.4 et %I0.0.5. Ces bits ne sont pas exclusivement réservés à ces blocs fonction. L'affectation de ces bits doit être déterminée selon l'utilisation de ces ressources dédiées par d'autres blocs fonction.

Illustration

L'illustration suivante présente un exemple de bloc fonction compteur rapide (FC) en mode mot simple.



Paramètres

Le tableau suivant présente les différents paramètres du bloc fonction compteur rapide (FC).

Paramètre	Etiquette	Description
Fonction	TYPE	Paramètre défini lors de la configuration et permettant de choisir entre le compteur et le décompteur.
Valeur de présélection	%FCi.P %FCi.PD	Valeur initiale définie : ->entre 1 et 65 535 en mode standard, ->entre 1 et 4 294 967 295 en mode mot double.
Réglable	O/N	Lorsqu'il est défini sur O, il est possible de modifier la valeur de présélection %FCi.P ou %FCi.PD et la valeur courante %FCi.V ou %FCi.VD à l'aide de l'afficheur ou de l'éditeur de tables d'animation. Lorsqu'il est défini sur N, il n'est pas possible d'accéder à cette présélection.
Valeur courante	%FCi.V %FCi.VD	La valeur courante évolue de manière croissante ou décroissante selon la fonction sélectionnée (comptage ou décomptage). Pour le comptage, la valeur de comptage courante est mise à jour. Elle peut atteindre 65 535 en mode standard (%FCi.V) et 4 294 967 295 en mode mot double (%FCi.VD). Pour le décomptage, la valeur courante est la valeur de présélection %FCi.P ou %FCi.PD. Elle peut décroître jusqu'à zéro.
Entrée pour valider	IN	A l'état 1, la valeur courante est mise à jour selon les impulsions appliquées à l'entrée physique. A l'état 0, la valeur courante reste inchangée.
Remise à zéro	%FCi.R	Paramètre utilisé pour initialiser le bloc. A l'état 1, la valeur courante est remise à 0 lorsque le bloc est configuré en tant que compteur, ou définie sur %FCi.P ou %FCi.PD lorsqu'il est configuré en tant que décompteur. Le bit Terminé %FCi.D reprend sa valeur par défaut.
Terminé	%FCi.D	Ce bit est réglé sur 1 lorsque %FCi.V ou %FCi.VD atteint %FCi.P ou %FCi.PD (bloc configuré en tant que compteur) ou lorsque %FCi.V ou %FCi.VD atteint zéro (bloc configuré en tant que décompteur). Ce bit en lecture seule est remis à 0 uniquement lorsque le paramètre %FCi.R est réglé sur 1.

Remarque

Lorsque le bloc est configuré comme réglable, l'application peut modifier la valeur de présélection %FCi.P ou %FCi.PD et la valeur courante %FCi.V ou %FCi.VD à tout moment. Cependant, une nouvelle valeur est prise en compte uniquement lorsque la réinitialisation de l'entrée est active ou sur le front montant de la sortie %FCi.D. Cela permet d'effectuer plusieurs comptages successifs sans perdre une seule impulsion.

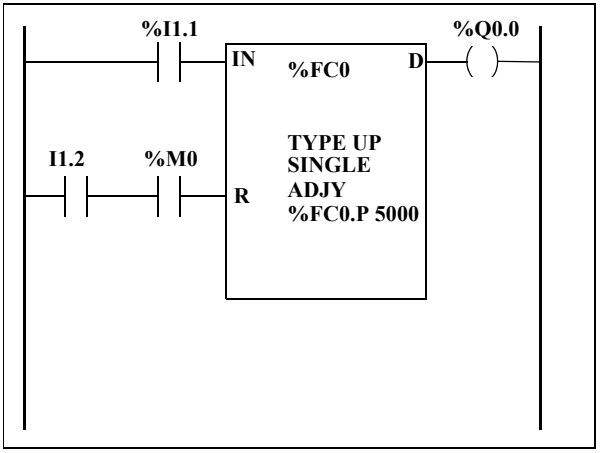
Fonctionnement

Lorsque le bloc est configuré comme compteur, la valeur courante est incrémentée de 1 dès qu'un front montant apparaît au niveau de l'entrée dédiée. Lorsque la valeur de présélection %FCi.P ou %FCi.PD est atteinte, le bit de sortie Terminé %FCi.D est mis à 1.

Lorsque le bloc est configuré comme décompteur, la valeur courante est diminuée de 1 dès qu'un front montant apparaît au niveau de l'entrée dédiée. Lorsque la valeur est zéro, le bit de sortie Terminé %FCi.D est mis à 1.

Configuration et programmation

Dans l'exemple ci-dessous, l'application compte le nombre d'éléments (5 000 maximum) pendant que %I1.1 est mis à 1. L'entrée pour %FC0 est l'entrée dédiée %I0.0.2. Lorsque la valeur de présélection est atteinte, %FC0.D est défini sur 1 et conserve la même valeur jusqu'à ce que %FC0.R soit commandé par le résultat de l'opération booléenne "AND" sur %I1.2 et %M0.




```
BLK      %FC0
LD       %I1.1
IN
LD       %I1.2
AND      %M0
R
OUT_BLK
LD D
ST %Q0.0
END_BLK
```

Cas particuliers

Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction %FC :

Cas spécifique	Description
Effet d'une reprise à froid (%S0=1)	Réinitialise les attributs %FC sur les valeurs configurées par l'utilisateur ou l'application utilisateur.
Effet d'une reprise à chaud (%S1=1)	Aucun effet.
Effet de l'arrêt de l'automate	%FC continue à compter selon les paramètres activés au moment de l'arrêt de l'automate.

Bloc fonction compteur rapide (%VFC)

Présentation

Le bloc fonction compteur rapide (%VFC) peut être configuré à l'aide de TwidoSuite pour exécuter l'une des fonctions suivantes :

- Compteur/décompteur
- Compteur/décompteur bi-phases
- Compteur simple
- Décompteur simple
- Fréquencemètre

Le bloc %VFC prend en charge le comptage des entrées TOR pour des fréquences allant jusqu'à 20 kHz en mode de calcul mot simple ou mot double. Les automates compacts TWDLC••40DRF peuvent contenir jusqu'à deux compteurs rapides (VFC) alors que les autres gammes d'automates compacts ne peuvent en comporter qu'un seul. Quant aux automates modulaires, ils peuvent en configurer jusqu'à deux.

Affectations des E/S dédiées

Les blocs fonction compteur rapide (%VFC) utilisent des entrées dédiées et des entrées et sorties auxiliaires. Ces entrées et ces sorties ne sont pas exclusivement réservées à ces blocs fonction. Leur affectation doit être déterminée selon l'utilisation de ces ressources dédiées par d'autres blocs fonction. Le tableau ci-après récapitule les affectations :

		Entrées principales		**Entrées auxiliaires**		**Sorties réflexes**	
%VFC0	Utilisation choisie	Entrée IA	Entrée IB	IPres	Ica	Sortie 0	Sortie 1
	Compteur/décompteur	%I0.0.1	%I0.0.0 (CO = 0/DE = 1)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Compteur/Décompteur bi-phases	%I0.0.1	%I0.0.0 (Impulsion)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Compteur simple	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Décompteur simple	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Fréquencemètre	%I0.0.1	(2)	(2)	(2)	(2)	(2)
%VFC1	Utilisation choisie	Entrée IA	Entrée IB)	IPres	Ica	Sortie 0	Sortie 1
	Compteur/décompteur	%I0.0.7	%I0.0.6 (CO = 0/DE = 1)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Compteur/Décompteur bi-phases	%I0.0.7	%I0.0.6 (Impulsion)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Compteur simple	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Décompteur simple	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Fréquencemètre	%I0.0.7	(2)	(2)	(2)	(2)	(2)

Commentaires:

(1) = facultatif

(2) = non utilisé

Ipres = entrée de présélection

Ica= entrée de capture

Entrée IA = entrée d'impulsion

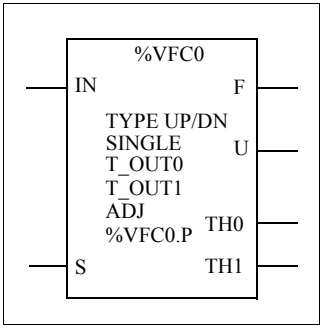
Entrée IB = impulsions ou UP/DO

UP/DO = Comptage / Décomptage

Lorsqu'elle n'est pas utilisée, l'entrée ou la sortie reste une E/S TOR normale disponible gérée par l'application dans le cycle principal.

Si %I0.0.2 est utilisé, %FC0 n'est pas disponible.
Si %I0.0.3 est utilisé, %FC2 n'est pas disponible.
Si %I0.0.4 est utilisé, %FC3 n'est pas disponible.

Illustration La figure suivante représente le compteur rapide (%VFC) en mode mot simple :



Caractéristiques Le tableau suivant répertorie les caractéristiques du bloc fonction compteur rapide (%VFC).

Fonction	Description	Valeurs	Utilisation du bloc %VFC	Accès en cours d'exécution
Valeur courante (%VFCi.V) (%VFCi.VD*)	La valeur courante est augmentée ou diminuée en fonction des entrées physiques et de la fonction sélectionnée. Cette valeur peut-être présélectionnée ou initialisée à l'aide de l'entrée de présélection (%VFCi.S).	%VFCi.V : 0 -> 65535 %VFCi.VD : 0 -> 4294967295	CM	Lecture
Valeur de présélection (%VFCi.P) (%VFCi.PD*)	Uniquement utilisée par la fonction compteur/décompteur et par le comptage ou le décomptage simple.	%VFCi.P : 0 -> 65535 %VFCi.PD : 0 -> 4294967295	CM ou FM	Lecture et écriture (1)
Valeur de capture (%VFCi.C) (%VFCi.CD*)	Uniquement utilisée par la fonction de comptage/ décomptage et par le comptage/décomptage simple.	%VFCi.C : 0 -> 65535 %VFCi.CD : 0 -> 4294967295	CM	Lecture
Sens de comptage (%VFCi.U)	Défini par le système, ce bit est utilisé par la fonction de comptage/décomptage pour indiquer le sens de comptage : Pour un compteur/décompteur simple, %I0.0.0 détermine le sens de %VFC0, et %I0.0.6 détermine le sens de %VFC1. Pour un compteur/décompteur bi-phases, la différence de phase entre les deux signaux détermine le sens de comptage. Pour %VFC0, %I0.0 est dédié à IB et %I0.1 à IA. Pour %VFC1, %I0.6 est dédié à IB et %I0.7 à IA.	0 (Décomptage) 1 (Comptage)	CM	Lecture

Fonction	Description	Valeurs	Utilisation du bloc %VFC	Accès en cours d'exécution
Activation de la sortie réflexe 0 (%VFCi.R)	Validation de la sortie réflexe 0	0 (Désactivé) 1 (Activé)	CM	Lecture et écriture (2)
Activer sortie réflexe 1 (%VFCi.S)	Validation sortie réflexe 1	0 (Désactivé) 1 (Activé)	CM	Lecture et écriture (2)
Valeur seuil S0 (%VFCi.S0) %VFCi.S0D	Contient la valeur de seuil 0. Sa signification est définie lors de la configuration du bloc fonction. Remarque : Cette valeur doit être inférieure à %VFCi.S1.	%VFCi.S0 : 0 -> 65535 %VFCi.S0D : 0 -> 4294967295	CM	Lecture et écriture (1)
Valeur seuil S1 (%VFCi.S1) (%VFCi.S1D*)	Contient la valeur de seuil 0. Sa signification est définie lors de la configuration du bloc fonction. Remarque : Cette valeur doit être supérieure à %VFCi.S0.	%VFCi.S1 : 0 -> 65535 %VFCi.S1D : 0 -> 4294967295	CM	Lecture et écriture (1)
Base temps de la mesure de fréquence (%VFCi.T)	Elément de configuration de la base temps (100 ou 1 000 millisecondes).	1 000 ou 100	MF	Lecture et écriture (1)
Réglable (Y/N)	Elément de configuration qui, lorsqu'il est sélectionné, permet à l'utilisateur de modifier les valeurs de présélection, de seuil et de base temps de la mesure de fréquence en cours d'exécution.	N (Non) Y (Oui)	CM ou FM	Non
Entrée pour valider (IN)	Utilisée pour valider ou inhiber la fonction courante.	0 (Non)	CM ou FM	Lecture et écriture (3)
Entrée de présélection (S)	Dépend de la configuration à l'état 1 : ● comptage/décomptage ou décomptage : initialise la valeur courante avec la valeur de présélection. ● comptage simple : remet la valeur courante à 0. Cette fonction permet également d'initialiser la commande des sorties seuils et prend en compte toutes les modifications apportées par un utilisateur aux valeurs seuils définies par l'afficheur ou le programme utilisateur.	0 ou 1	MC ou MF	Lecture et écriture
Sortie débordement (F)	de 0 à 65 535 ou de 65 535 à 0 en mode standard de 0 à 4 294 967 295 ou de 4 294 967 295 à 0 en mode mot double	0 ou 1	CM	Lecture

Fonction	Description	Valeurs	Utilisation du bloc %VFC	Accès en cours d'exécution
Seuil Bit 0 (%VFCi.TH0)	A l'état 1 lorsque la valeur courante est supérieure ou égale à la valeur seuil %VFCi.S0. Nous conseillons de tester ce bit une seule fois dans le programme, car il est mis à jour en temps réel. L'application utilisateur est responsable de la validité de la valeur au moment de son utilisation.	0 ou 1	CM	Lecture
Seuil Bit 1 (%VFCi.TH1)	A l'état 1 lorsque la valeur courante est supérieure ou égale à la valeur seuil %VFCi.S1. Nous conseillons de tester ce bit une seule fois dans le programme, car il est mis à jour en temps réel. L'application utilisateur est responsable de la validité de la valeur au moment de son utilisation.	0 ou 1	MC	Lecture

(*)Correspond à une variable de double mot 32 bits. L'option de double mot est disponible sur tous les automates à l'exception des automates Twido TWDLC•A10DRF.

(1) Accessible en écriture uniquement si la fonction Réglable est réglée sur un.

(2) Accès disponible si configuré uniquement.

(3)Accès en lecture et en écriture seulement à partir de l'application. Accès impossible à partir de l'afficheur ou de l'éditeur de tables d'animation.

MC = Mode Comptage

MF = Mode Fréquencemètre

Description de la fonction de comptage

La fonction de comptage très rapide (%VFC) fonctionne à une fréquence maximale de 20 kHz et pour une plage de valeurs allant de 0 à 65 535 en mode standard et de 0 à 4 294 967 295. Les impulsions de comptage sont appliquées de la manière suivante.

Tableau :

Fonction	Description	%VFC0		%VFC1	
		IA	IB	IA	IB
Compteur/ Décompteur	Les impulsions sont appliquées à l'entrée physique ; l'opération courante (comptage/décomptage) est définie par l'état de l'entrée physique IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Compteur/ Décompteur bi-phases	Les deux phases du codeur sont appliquées aux entrées physiques IA et IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Compteur simple	Les impulsions sont appliquées à l'entrée physique IA. IB n'est pas utilisée.	%I0.0.1	ND	%I0.0.7	ND
Décompteur simple	Les impulsions sont appliquées à l'entrée physique IA. IB n'est pas utilisée.	%I0.0.1	ND	%I0.0.7	ND

Remarques sur les blocs fonction

Les opérations de comptage ou de décomptage sont effectuées sur le front montant des impulsions et ce, uniquement lorsque le bloc compteur est activé.

Deux entrées facultatives sont utilisées en mode de comptage : ICa et IPres. ICa est utilisée pour capturer la valeur courante (%VFCi.V ou %VFCi.VD) et la stocker dans %VFCi.C ou %VFCi.CD. Les entrées ICa sont définies sur %I0.0.3 pour %VFC0 et sur %I0.0.4 pour %VFC1, le cas échant.

Lorsque l'entrée IPres est active, la valeur courante est affectée de la manière suivante :

- Pour le comptage, %VFCi.V ou %VFCi.VD sont remis à 0.
- Pour le décomptage, %VFCi.V ou %VFCi.VD sont écrits respectivement avec le contenu de %VFCi.P ou %VFCi.PD.
- Pour le comptage de fréquence, %VFCi.V ou %VFCi.PD sont mis à 0.

Avertissement : %VFCi.F sera également mis à 0. Les entrées IPres sont définies sur %I0.0.2 pour %VFC0 et sur %I0.0.5 pour %VFC1 si cette valeur est disponible.

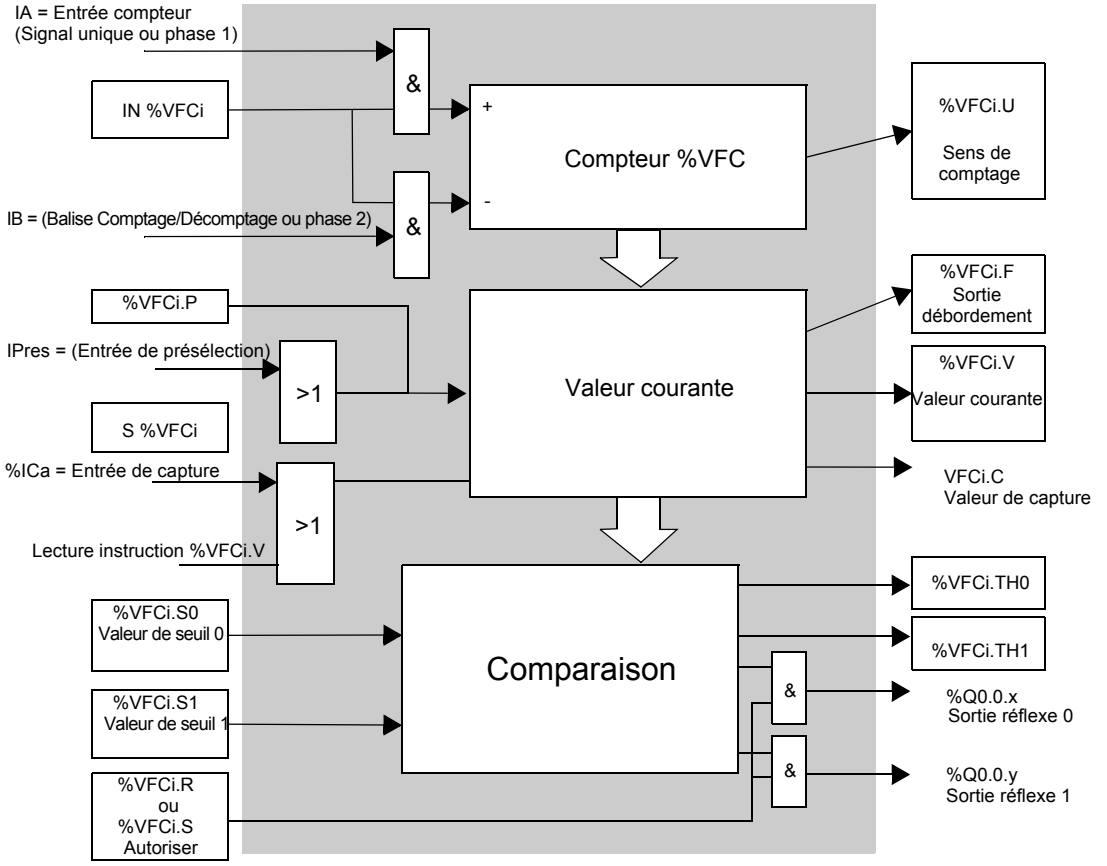
Remarques sur les sorties des blocs fonction

Pour toutes les fonctions, les valeurs courantes sont comparées aux deux seuils (%VFCi.S0 ou %VFCi.S0D et %VFCi.S1 ou %VFCi.S1D). Les deux objets bits (%VFCi.TH0 et %VFCi.TH1) sont fonction des résultats de cette comparaison. C'est-à-dire qu'ils sont réglés sur 1 lorsque la valeur courante est supérieure ou égale au seuil correspondant ou remis à 0 dans le cas contraire. Les sorties réflexes (si elles sont configurées) sont réglées sur 1 en fonction de ces comparaisons. Remarque : Aucune, une ou deux sorties peuvent être configurées.

%VFC.U est une sortie du bloc fonction. Elle indique le sens de variation du compteur (1 pour comptage, 0 pour décomptage).

Schéma de la fonction de comptage

L'illustration suivante représente un schéma de fonction de comptage en mode standard (en mode double mot, vous utiliserez en conséquence les variables de la fonction de doubles mots) :



Note : Les sorties sont gérées indépendamment du temps de cycle automate. Le temps de réponse est compris entre 0 et 1 ms.

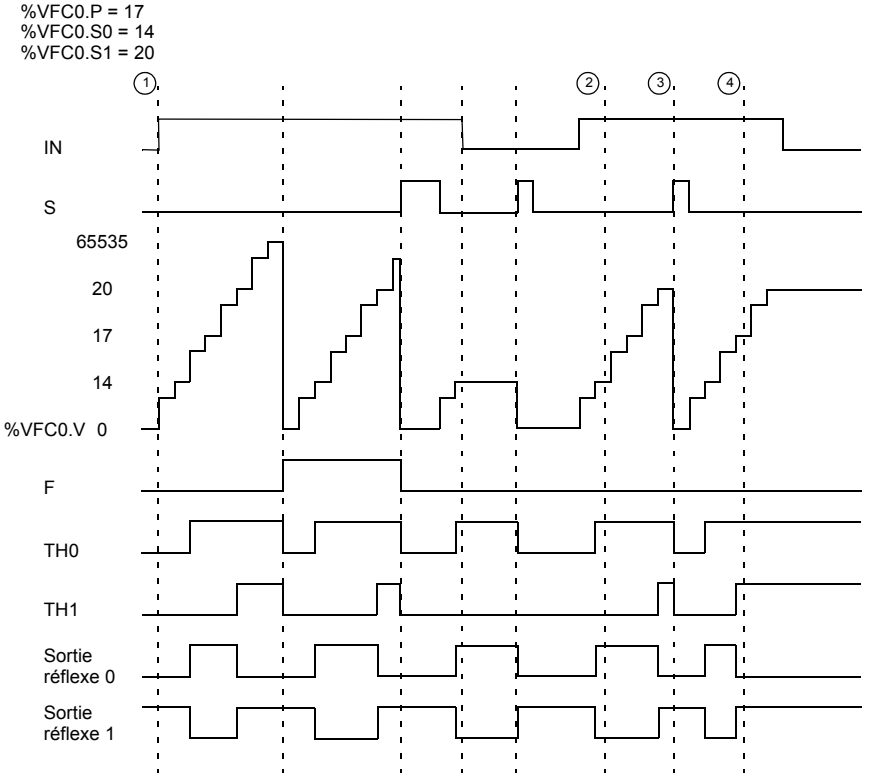
Opération de comptage simple

Voici un exemple de l'utilisation de %VFC en mode comptage simple. Les éléments de configuration suivants ont été définis pour cet exemple :

La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	valeur < %VFC.S0	%VFC0.S0 <= valeur < %VFC0.S1	valeur >= %VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X

Exemple de chronogramme :



(1) : %VFC0.U = 1 car %VFC est un compteur

(2) : modification de %VFC0.S1 sur 17

(3) : l'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du comptage suivant

(4) : une interception de la valeur courante a lieu, ainsi, %VFC0.C = 17

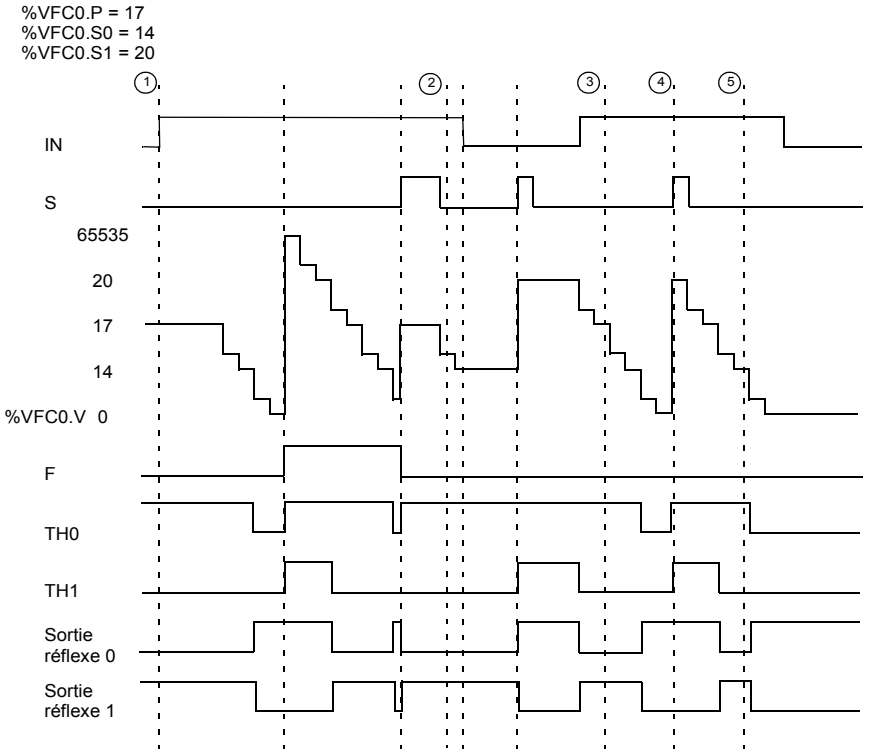
Opération de décomptage simple

Voici un exemple de l'utilisation de %VFC en mode décomptage simple. Les éléments de configuration suivants ont été définis pour cet exemple :

La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	valeur < %VFC.S0	%VFC0.S0 <= valeur < %VFC0.S1	valeur >= %VFC0.S1
%Q0.0.2	X		X
%Q0.0.3		X	

Exemple :



(1) : %VFC0.U = 0 car %VFC est un décompteur

(2) : modification de %VFC0.P sur 20

(3) : modification de %VFC0.S1 sur 17

(4) : L'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du décompte suivant

(5) : une capture de la valeur courante a lieu, ainsi, %VFC0.C = 17

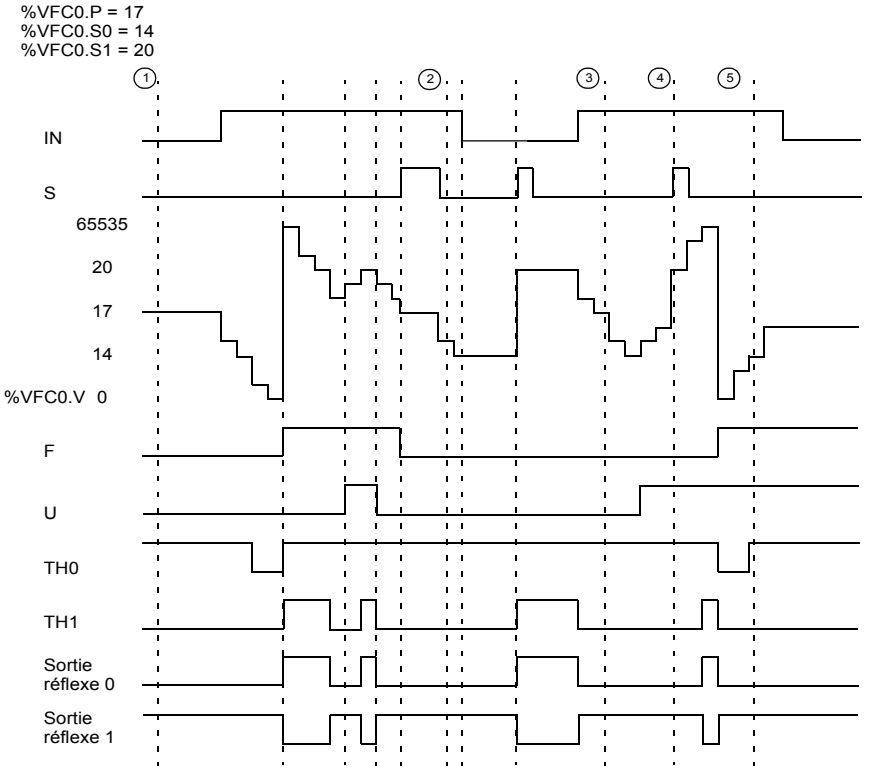
Opération de comptage/ décomptage

Voici un exemple de l'utilisation de %VFC en mode comptage/décomptage. Les éléments de configuration suivants ont été définis pour cet exemple :

La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	valeur < %VFC.S0	%VFC0.S0 <= valeur < %VFC0.S1	valeur >= %VFC0.S1
%Q0.0.2			X
%Q0.0.3	X	X	

Exemple :



① : entrées IN et S mises à 1

② : modification de %VFC0.P sur 20

③ : modification de %VFC0.S1 sur 17

④ : L'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du décompte suivant

⑤ : une capture de la valeur courante a lieu, ainsi, %VFC0.C = 17

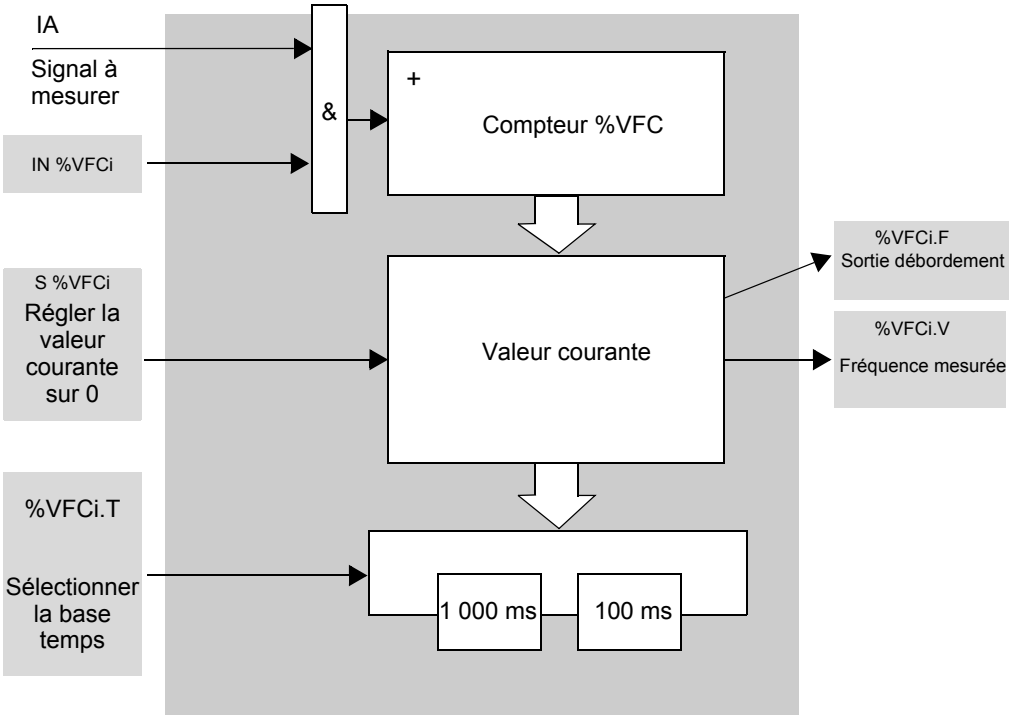
Description de la fonction Fréquencemètre

La fonction Fréquencemètre d'un %VFC est utilisée pour mesurer la fréquence en Hz d'un signal périodique sur l'entrée IA. La plage de fréquences pouvant être mesurées est comprise entre 10 et 20 kHz. L'utilisateur peut choisir entre deux bases temps. Ce choix est effectué via un nouvel objet %VFC.T (Base temps). Une valeur de 100 correspond à une base temps de 100 ms et une valeur de 1 000 correspond à une base temps d'une seconde.

Base temps	Plage de mesure	Précision	Mise à jour
100 ms	100 Hz à 20 kHz	0,05 % pour 20 kHz, 10 % pour 100 Hz	10 fois par seconde
1 s	10 Hz à 20 kHz	0,005 % pour 20 kHz, 10 % pour 10 Hz	Une fois par seconde

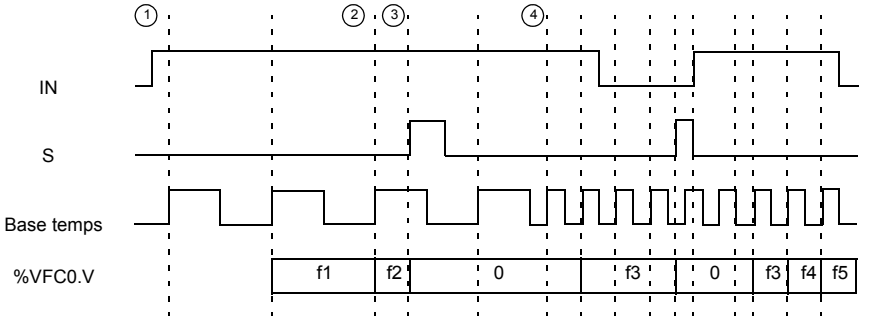
Schéma de la fonction Fréquencemètre

Exemple de schéma de fonction Fréquencemètre :



Opération Fréquencemètre

Voici un exemple de chronogramme de l'utilisation de %VFC en mode Fréquencemètre :



① : la mesure de la première fréquence débute ici

② : la valeur de la fréquence courante est mise à jour

③ : entrées IN et S mises à 1

④ : modification de %VFC0.T sur 100 ms : cette modification annule la mesure en cours et en lance une autre.

Cas particuliers

Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction %VFC.

Cas spécifique	Description
Effet d'une reprise à froid (%S0=1)	Utilise les valeurs configurées par l'utilisateur ou par l'application utilisateur pour régler tous les attributs %VFC.
Effet d'une reprise à chaud (%S1=1)	Aucun effet
Effet de l'arrêt de l'automate	Le %VFC s'arrête et les sorties maintiennent leur état courant.

Emission/réception de messages – Instruction d'échange (EXCH)

Présentation

Il est possible de configurer un automate Twido afin qu'il puisse communiquer avec des périphériques esclaves Modbus ou envoyer et/ou recevoir des messages en mode ASCII (mode caractères).

TwidoSuite propose les fonctions suivantes pour ces communications :

- Instruction EXCH pour l'émission/la réception de messages
- Bloc fonction de contrôle d'échange %MSG assurant le contrôle des échanges de données

L'automate Twido utilise le protocole configuré pour le port spécifié lors du traitement d'une instruction EXCH. Il est possible d'affecter un protocole différent à chaque port de communication. Pour accéder aux ports de communication, ajoutez le numéro de port à la fonction EXCH ou %MSG (EXCH1, EXCH2, %MSG1, %MSG2).

De plus, les automates TWDLC•E40DRF implémentent la messagerie Modbus TCP sur le réseau Ethernet à l'aide de l'instruction EXCH3 et de la fonction %MSG3.

Instruction EXCH

L'instruction EXCH permet à un automate Twido d'envoyer et/ou recevoir des informations vers/depuis des périphériques ASCII. L'utilisateur définit une table de mots (%MWi:L) contenant les données à envoyer et/ou recevoir (jusqu'à 250 octets de données en émission et/ou réception). Le format des tables de mots fait l'objet d'une description dans les sections relatives à chaque protocole. Un échange de message est exécuté à l'aide de l'instruction EXCH.

Syntaxe

La syntaxe à utiliser pour l'instruction EXCH est la suivante :

[EXCHx %MWi:L]

Où : x = numéro de port série (1 ou 2), x = port Ethernet (3), L = nombre total de mots de la table de mots (121 maximum). Les valeurs contenues dans la table de mots interne %MWi:L prennent la forme i+L <= 255.

L'automate Twido doit terminer l'échange ordonné par la première instruction EXCHx avant qu'une nouvelle instruction d'échange puisse être lancée. Le bloc fonction %MSG doit être utilisé lors de l'envoi de plusieurs messages.

Note : Pour plus d'informations sur l'instruction EXCH3 de messagerie Modbus TCP, voir (Messagerie Modbus TCP, Twido, Guide de programmation).

Bloc fonction de contrôle d'échange (%MSGx)

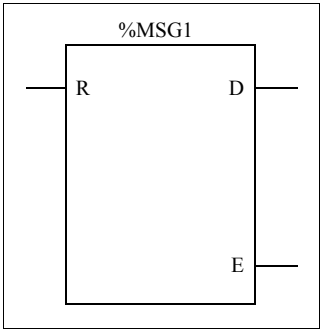
Introduction

Note : Le "x" de %MSGx désigne le port de l'automate : "x = 1 ou 2"
- x = 1 ou 2 correspond respectivement au port série 1 ou 2 de l'automate ;
- X = 3 correspond au port réseau Ethernet de l'automate (sur l'automate TWDLC•E40DRF uniquement). Pour plus d'informations sur la fonction %MSG3, voir (Messagerie Modbus TCP, Twido, Guide de programmation).

Le bloc fonction %MSGx assure la gestion des échanges de données. Ce bloc a trois fonctions :
- Vérification des erreurs de communication
  Cette fonction a pour but de s'assurer que la longueur du bloc (table de mots) programmée avec l'instruction EXCH est suffisante pour le stockage du message à envoyer (comparaison de la longueur programmée dans l'octet de poids faible du premier mot de la table de mots).
  Erreur 1 : en cas de commande erronée, de table configurée de manière incorrecte, de mauvais caractère reçu (vitesse, parité, etc.) ou de saturation de la table de réception (non mise à jour).
- Coordination de plusieurs messages
  Afin d'assurer la coordination de l'envoi de plusieurs messages, le bloc fonction %MSGx contient des informations permettant de s'assurer que l'émission de chaque message est bien terminée.
- Emission de messages prioritaires
  Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

La programmation du bloc fonction %MSGx est facultative.

Illustration L'exemple suivant illustre le bloc fonction %MSGx.



Paramètres Le tableau suivant présente les différents paramètres du bloc fonction %MSGx.

Paramètre	Etiquette	Valeur
Entrée (ou instruction) RAZ	R	A l'état 1, réinitialise la communication : %MSGx.E = 0 et %MSGx.D = 1.
Sortie Communication terminée	%MSGx.D	A l'état 1, la communication est terminée si : ● fin d'émission (si émission) ● fin de réception (réception du caractère de fin) ● Erreur ● réinitialisation du bloc A l'état 0, une requête est en cours.
Sortie Défaut (Erreur)	%MSGx.E	A l'état 1, la communication est terminée si : ● commande incorrecte ● table configurée de manière incorrecte ● réception d'un caractère incorrect (vitesse, parité, etc.) ● table de réception pleine (non mise à jour) l'état 0, la longueur du message et la liaison sont correctes.

Si une erreur survient lors de l'exécution d'une instruction EXCH, les bits %MSGx.D et %MSGx.E sont réglés sur 1, le mot système %SW63 contient le code de l'erreur du Port 1 et le mot système %SW64 celui du Port 2. Voir *p. 683*.

Entrée RAZ (R)

Lorsque l'entrée RAZ est mise à 1 :

- L'émission de tous les messages est interrompue.
- La sortie Défaut (Erreur) est remise à 0.
- Le bit Terminé est mis à 1.

Un nouveau message peut être envoyé.

Sortie Défaut (Erreur) (%MSGx.E)

La sortie Défaut est mise à 1 en cas d'erreur de programmation des communications ou d'erreur d'émission d'un message. La sortie Défaut est mise à 1 si le nombre d'octets définis dans le bloc de données associé à l'instruction EXCH (mot 1, octet de poids faible) est supérieur à 128 (+80 en hexadécimal par FA).

La sortie Défaut est également mise à 1 en cas de problème lors de l'envoi d'un message Modbus vers un périphérique Modbus. Dans ce cas, l'utilisateur devra vérifier la connexion et s'assurer que le périphérique de destination peut recevoir des communications Modbus.

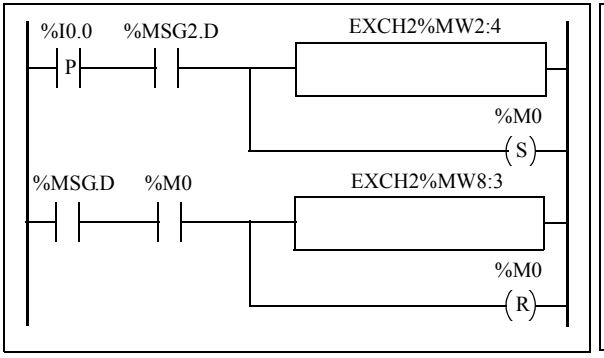
Sortie Communication terminée (%MSGx.D)

Lorsque la sortie Communication terminée est mise à 1, l'automate Twido est prêt à envoyer un autre message. L'utilisation de la sortie %MSGx.D est recommandée en cas d'envoi de plusieurs messages. Si cette sortie n'est pas utilisée, les messages pourront être perdus.

Emission de plusieurs messages successifs

L'exécution de l'instruction EXCH permet d'activer un bloc message dans le programme d'application. Le message est émis si le bloc message n'est pas déjà actif (%MSGx.D = 1). Lorsque plusieurs messages sont envoyés au cours du même cycle, seul le premier message est émis. La gestion de l'émission de plusieurs messages à l'aide du programme incombe à l'utilisateur.

Exemple d'émission de deux messages successifs sur le port 2 :

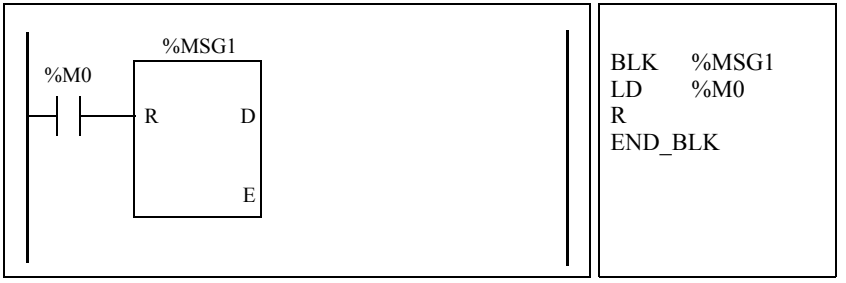



```
LDR     %I0.0
AND     %MSG2.D
[EXCH2 %MW2:4]
S       %M0
LD      %MSG2.D
AND     %M0
[EXCH2 %MW8:3]
R       %M0
```

Réinitialisation des échanges

L'annulation d'un échange survient lors de l'activation de l'entrée (ou de l'instruction) R. Cette entrée initialise la communication, remet à zéro la sortie %MSGx.E et met la sortie %MSGx.D à 1. Notez qu'il est possible de réinitialiser une communication si une défaillance est détectée.

Exemple de réinitialisation d'un échange :



Cas particuliers

Le tableau présente les cas particuliers de fonctionnement du bloc fonction %MSGx.

Cas particulier	Description
Effet d'un redémarrage à froid (%S0=1)	Force la réinitialisation de la communication.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet
Effet d'un arrêt de l'automate	Si un message est en cours d'émission, l'automate interrompt le transfert et réinitialise les sorties %MSGx.D et %MSGx.E.

18.2 Fonctions horodateur

Présentation

Objet de ce sous-chapitre

Ce sous chapitre offre une description des fonctions de gestion du temps des automates Twido.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonctions horloges	546
Blocs horodateurs	547
Horodatage	549
Réglage de la date et de l'heure	551

Fonctions horloges

Introduction

Les automates Twido possèdent une fonction Date/Heure. Cette fonction requiert l'option Horodateur (RTC) et permet d'utiliser :

- des **blocs horodateurs**, pour la programmation d'actions à des moments prédéfinis ou calculés.
- une fonctionnalité d'**horodatage**, pour la consignation des durées et des calendriers d'événements et la mesure de la durée de ces derniers.

Pour accéder à l'horloge Date/Heure Twido, sélectionnez **Blocs horodateurs** dans la tâche TwidoSuite **Programme** → **Configurer** → **Configurer les données**. Notez que cette horloge peut également être réglée à l'aide d'un programme. En cas d'extinction de l'automate, les réglages de l'horloge sont conservés en mémoire pendant un maximum de 30 jours, si la batterie de l'automate était en charge pendant les six heures qui ont précédé l'extinction de l'automate.

L'affichage de l'horloge Date/Heure se fait au format "24 heures" et tient compte des années bissextiles.

Valeur de correction RTC

La définition de la valeur de correction de l'horodateur est nécessaire à son bon fonctionnement. Chaque horodateur possède sa propre valeur de correction, figurant au sein même de l'unité. Pour configurer cette valeur dans TwidoSuite, sélectionnez l'option **Configurer RTC** de l'**Utilitaire de surveillance TwidoSuite** accessible via le **Lanceur d'applications TwidoSuite**.

Blocs horodateurs

Introduction

Les blocs horodateurs permettent de programmer et de contrôler des actions selon un calendrier précis (mois, jour et heure). Un maximum de 16 blocs horodateurs peuvent être programmés. Ces blocs ne requièrent aucune saisie programme.

Note : Vérifiez le bit système %S51 et le mot système %SW118 afin de vous assurer que l'option horodateur (RTC) est installée. Reportez-vous à la rubrique *p. 674*. L'option RTC est requise pour l'utilisation de blocs horodateurs.

Paramètres

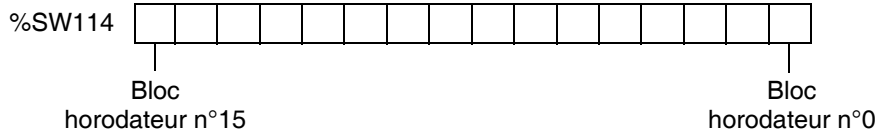
Le tableau suivant répertorie les paramètres d'un bloc horodateur :

Paramètre	Format	Fonction/Plage
Numéro du bloc horodateur	n	n = 0 à 15
Configuré	Case à cocher	Cochez cette case pour configurer le bloc horodateur sélectionné.
Bit de sortie	%Qx.y.z	L'affectation de la sortie est activée par le bloc horodateur : %Mi ou %Qj.k. Cette sortie est mise à 1 lorsque les paramètres de date et d'heure courants sont compris entre les paramètres de début et de fin de la période active.
Mois de début	janvier à décembre	Mois au cours duquel débute le bloc horodateur.
Mois de fin	janvier à décembre	Mois au cours duquel s'achève le bloc horodateur.
Date de début	1 - 31	Jour au cours duquel débute le bloc horodateur.
Date de fin	1 - 31	Jour au cours duquel s'achève le bloc horodateur.
Heure de début	hh:min	Heure à laquelle débute le bloc horodateur. Définie par l'heure (0 à 23), suivie des minutes (0 à 59).
Heure d'arrêt	hh:min	Heure à laquelle s'achève le bloc horodateur. Définie par l'heure (0 à 23), suivie des minutes (0 à 59).
Jour de la semaine	lundi à dimanche	Cases à cocher permettant de définir les jours au cours desquels sera activé le bloc horodateur.

Activation de blocs horodateurs

Les bits du mot système %SW114 activent (lorsqu'ils sont mis à 1) ou désactivent (lorsqu'ils sont mis à 0) le fonctionnement des 16 blocs horodateurs.

Affectation des blocs horodateurs dans %SW114 :



Par défaut (ou après un démarrage à froid), tous les bits de ce mot système sont mis à 1. L'utilisation de ces bits par le programme est optionnelle.

Sortie des blocs horodateurs

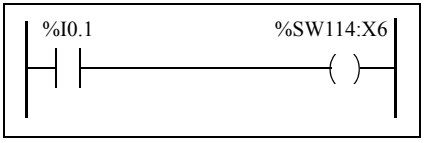
Si la même sortie (%Mi ou %Qj.k) est affectée par plusieurs blocs, c'est le OU des résultats de chacun des blocs qui est finalement affecté à cet objet (notez que la même sortie peut disposer de plusieurs « plages de fonctionnement »).

Exemple

Le tableau suivant présente les paramètres d'un programme d'arrosage pendant la période d'été:

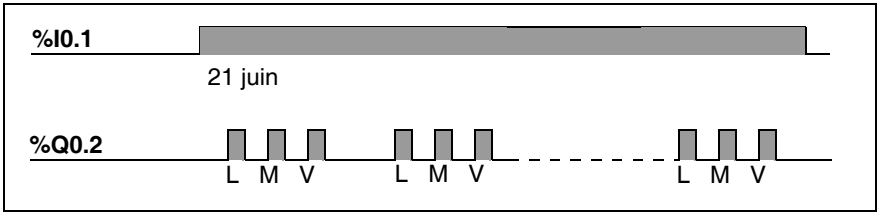
Paramètre	Valeur	Description
Bloc horodateur	6	Bloc horodateur numéro 6
Bit de sortie	%Q0.2	Activer la sortie %Q0.2
Mois de début	Juin	Débuter l'activité en juin
Mois de fin	Septembre	Arrêter l'activité en septembre
Date de début	21	Débuter l'activité le 21ème jour de juin
Date de fin	21	Arrêter l'activité le 21ème jour de septembre
Jour de la semaine	lundi, mercredi, vendredi	Exécuter l'activité les lundis, mercredis et vendredis
Heure de début	21:00	Débuter l'activité à 21:00
Heure d'arrêt	22:00	Arrêter l'activité à 22:00

Le programme suivant permet de désactiver le bloc horodateur grâce à un commutateur ou un détecteur d'humidité lié à l'entrée %I0.1.




```
LD    %I0.1
ST    %SW114:X6
```

Le chronogramme suivant illustre l'activation de la sortie %Q0.2.



Gestion de plage horaire par programme

Les paramètres de date et d'heure sont disponibles dans les mots système %SW50 à %SW53 (voir la rubrique *p. 683*). Il est ainsi possible d'effectuer un horodatage dans le programme de l'automate en effectuant des comparaisons arithmétiques entre la date et l'heure courantes et les valeurs immédiates ou les mots %MWi (ou %KWi), qui peuvent contenir des consignes.

Horodatage

Introduction

Les mots système %SW49 à %SW53 contiennent les paramètres de date et d'heure au format BCD (reportez-vous à la section *p. 490*), qui est utile pour l'affichage sur un périphérique ou la transmission vers ce périphérique. Ces mots système peuvent être utilisés pour stocker les paramètres de date et d'heure d'un événement (voir *p. 683*).

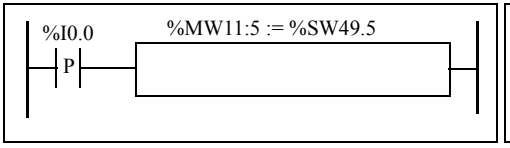
Note : Les paramètres de date et d'heure peuvent également être réglés à l'aide de l'afficheur optionnel (voir *p. 380*).

Datage d'un événement

Pour dater un événement, il suffit d'utiliser des opérations d'affectation, pour transférer le contenu de mots système vers des mots internes et de traiter ces mots internes (par exemple, la transmission vers l'afficheur à l'aide de l'instruction EXCH).

Exemple de programmation

L'exemple suivant montre comment dater un front montant sur l'entrée %I0.1.




```
LDR     %I0.0
[%MW11:5 := %SW49:5]
```

Dès qu'un événement est détecté, la table de mots contient :

Codage	Octet de poids fort	Octet de poids faible
%MW11		Jour de la semaine[1]
%MW12	00	Seconde
%MW13	Heure	Minute
%MW14	Mois	Jour
%MW15	Siècle	Année

Note : (1) 1 = lundi, 2 = mardi, 3 = mercredi, 4 = jeudi, 5 = vendredi, 6 = samedi, 7 = dimanche.

Exemple de table de mots

Exemple de données pour le lundi 19 avril 2002, à 13:40:30 :

Mot	Valeur (hex.)	Signification
%MW11	0001	Lundi
%MW12	0030	30 secondes
%MW13	1340	13 heures, 40 minutes
%MW14	0419	04 = avril, le 19
%MW15	2002	2002

Date et heure du dernier arrêt

Les mots système %SW54 à %SW57 contiennent les paramètres de date et d'heure du dernier arrêt et le mot %SW58 contient le code affichant la cause du dernier arrêt, au format BCD (voir *p. 683*).

Réglage de la date et de l'heure

Introduction

Pour mettre à jour les paramètres de date et d'heure, vous pouvez utiliser l'une des méthodes suivantes :

- TwidoSuite
  Utilisez la boîte de dialogue **Réglage de l'heure**. Cette boîte de dialogue est disponible :
  - à partir de l'**Utilitaire de surveillance TwidoSuite**, accessible via le **Lanceur d'applications TwidoSuite** ou
  - en sélectionnant **Programme** → **Mise au point** → **Contrôler l'automate**.
- Mots système
  Utilisez les mots système %SW49 à %SW53 ou le mot système %SW59.

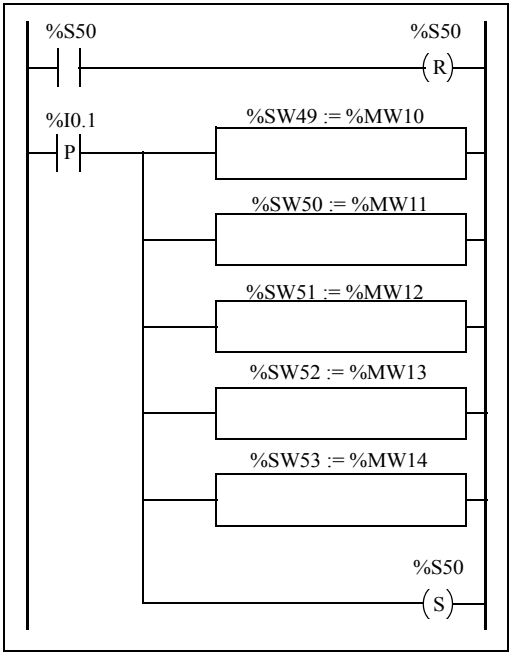
Les paramètres de date et d'heure peuvent être mis à jour uniquement lorsque la cartouche optionnelle de l'horodateur (TWDXCPRTC) est installée sur l'automate. Notez que les automates compacts de la série TWDLC••40DRF et les automates Twido Extreme TWDLEDCK1 disposent d'un RTC intégré.

Utilisation des mots %SW49 à %SW53

Pour utiliser les mots système %SW49 à %SW53 afin de régler la date et l'heure, le bit %S50 doit être réglé sur 1. Ce réglage a les conséquences suivantes :

- l'annulation de la mise à jour des mots %SW49 à %SW53 via l'horloge interne ;
- le transfert des valeurs écrites dans les mots %SW49 à %SW53 vers l'horloge interne.

Exemple de programmation :




```
LD      %S50
R       %S50

LDR     %I0.1
[%SW49 := %MW10]
[%SW50 := %MW11]
[%SW51 := %MW12]
[%SW52 := %MW13]
[%SW53 := %MW14]
S       %S50
```

Les mots %MW10 à %MW14 contiendront les nouveaux paramètres de date et d'heure au format BCD (reportez-vous à la section *p. 490*) et correspondront au codage des mots %SW49 à %SW53.

La table de mots doit contenir les nouveaux paramètres de date et d'heure :

Codage	Octet de poids fort	Octet de poids faible
%MW10		Jour de la semaine[1]
%MW11		Seconde
%MW12	Heure	Minute
%MW13	Mois	Jour
%MW14	Siècle	Année

Note : (1) 1 = lundi, 2 = mardi, 3 = mercredi, 4 = jeudi, 5 = vendredi, 6 = samedi, 7 = dimanche.

Exemple pour le lundi 19 avril 2002 :

Mot	Valeur (hex.)	Signification
%MW10	0001	Lundi
%MW11	0030	30 secondes
%MW12	1340	13 heures, 40 minutes
%MW13	0419	04 = avril, le 19
%MW14	2002	2002

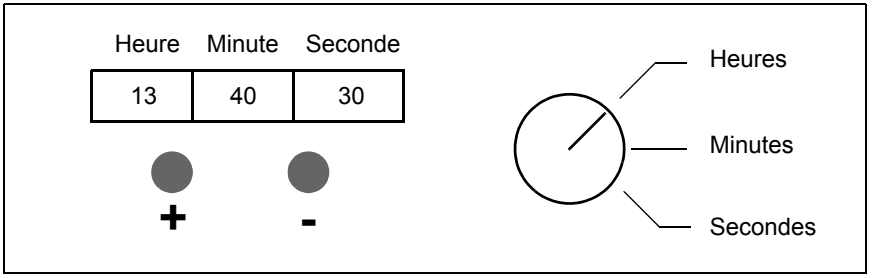
Utilisation du mot %SW59

Pour mettre à jour la date et l'heure, vous pouvez également utiliser le bit système %S59 et le mot système %SW59 de réglage de la date.

La mise du bit %S59 à 1 permet de régler les paramètres de date et d'heure courants à l'aide du mot %SW59 (reportez-vous à la section *p. 683*). Le mot système %SW59 permet d'incrémenter ou de décrémenter chacun des composants de date et d'heure sur un front montant.

Exemple de mise en œuvre

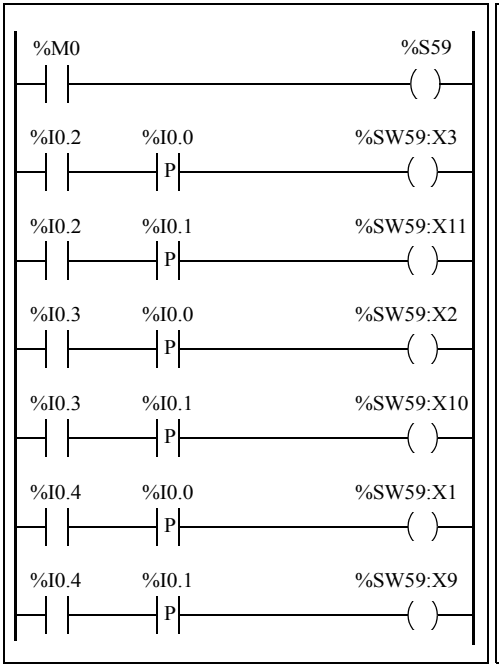
Le panneau avant ci-dessous permet de modifier le réglage de l'horloge interne (heures, minutes et secondes).



Description des commandes :

- Le commutateur Heures/Minutes/Secondes permet de sélectionner l'heure à modifier, respectivement à l'aide des entrées %I0.2, %I0.3 et %I0.4.
- Le bouton + permet d'incrémenter l'affichage de l'heure sélectionnée, à l'aide de l'entrée %I0.0.
- Le bouton - permet de décrémenter l'affichage de l'heure sélectionnée, à l'aide de l'entrée %I0.1.

Le programme suivant lit les entrées du panneau et règle l'horloge interne.




```
LD      %M0
ST      %S59
LD      %I0.2           (Heure)
ANDR    %I0.0
ST      %SW59:X3
LD      %I0.2
ANDR    %I0.1
ST      %SW59:X11
LD      %I0.3           (Minute)
ANDR    %I0.0
ST      %SW59:X2
LD      %I0.3
ANDR    %I0.1
ST      %SW59:X10
LD      %I0.4           (Seconde)
ANDR    %I0.0
ST      %SW59:X1
LD      %I0.4
ANDR    %I0.1
ST      %SW59:X9
```

18.3 Guide de démarrage rapide de l'automate PID Twido

Présentation

Vue d'ensemble

Cette section contient des informations sur le démarrage rapide avec le contrôle PID et les fonctions d'auto tuning disponibles sur les automates Twido.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Objectif du document	555
Etape 1 - Configuration des voies analogiques utilisées pour le contrôle	557
Etape 2 - Conditions préalables à la configuration de l'automate PID	559
Etape 3 - Configuration du PID	561
Etape 4 - Initialisation de la configuration du contrôle	567
Etape 5 - Fonction AT + PID pour l'installation du contrôle	571
Etape 6 - Mise au point des réglages	575

Objectif du document

Introduction

Ce guide de démarrage rapide a pour objectif de vous guider, à l'aide d'exemples pratiques, dans les étapes de configuration et d'installation des fonctions de contrôle de votre automate PID Twido.

Note : Mettre en place la fonction PID sur un automate Twido ne demande pas de connaissances particulières mais nécessite un certain degré de rigueur pour obtenir les meilleurs résultats possibles dans un délai le plus bref.

Contenu du document

Ce document explique les étapes suivantes :

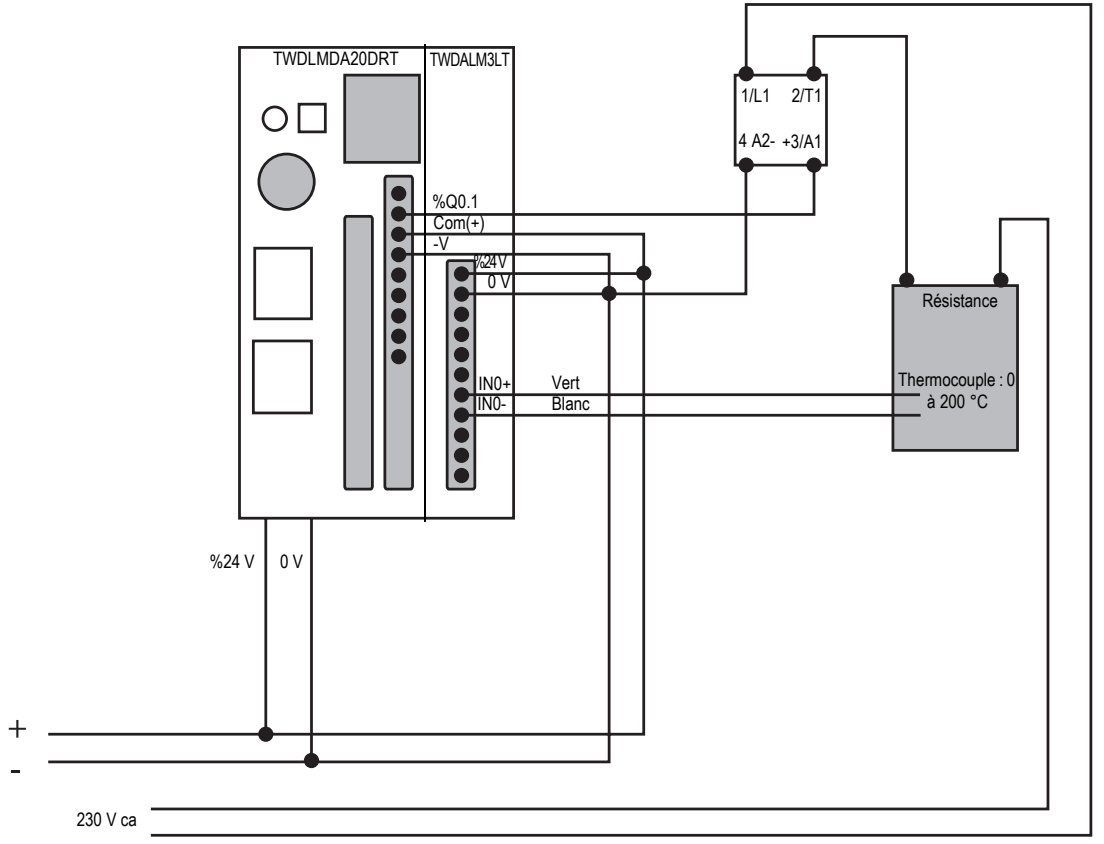
Etape	Description
1	Configuration des voies analogiques utilisées pour le contrôle
2	Conditions préalables à la configuration de l'automate PID
3	Configuration de l'automate PID
4	Initialisation de l'installation du contrôle
5	Installation du contrôle AT + PID
6	Mise au point et réglages

Exemple utilisé dans ce guide

Pour cet exemple, nous avons choisi un thermocouple de type K (0-200 °C).

Nous utiliserons un contrôle par transistor avec une sortie de la base automate contrôlée directement par l'automate PID à l'aide de PWM (voir *p. 561*).

Le schéma ci-dessous représente l'installation expérimentale utilisée dans l'exemple :



Etape 1 - Configuration des voies analogiques utilisées pour le contrôle

Présentation

Un automate PID utilise généralement un signal de retour analogique (appelé "mesure") pour mesurer la valeur à régler.

Cette valeur peut être un niveau, une température, une distance ou une autre valeur pour d'autres applications.

Exemple de signal de mesure analogique

Prenons l'exemple d'une mesure de température.

Le capteur envoie une mesure analogique qui dépend de la valeur mesurée renvoyée à l'automate. Pour une température et avec des capteurs de type PT100 ou thermocouple, le signal mesuré augmente avec la montée de la température courante.

Ajout d'une carte analogique (module d'expansion)

En mode local, une fois que vous avez sélectionné la base automate, ajoutez la carte analogique comme une extension de base. La numérotation des voies dépend de son emplacement de configuration.

Configuration de voies d'entrée analogiques

Le tableau suivant décrit la procédure de configuration des voies analogiques du module d'expansion :

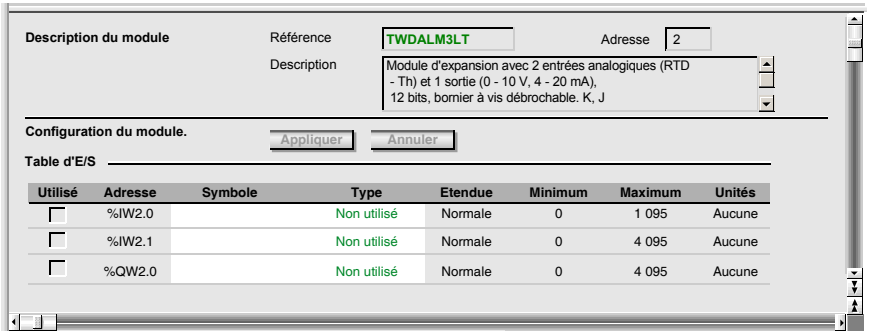
Etape	Action
1	Sélectionnez **Description** dans l'interface TwidoSuite. Reportez-vous à la section (Description des applications, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
2	Affichez le catalogue des produits et sélectionnez un module à ajouter à la description du système. Par exemple, **TWDALM3LT** pour mesurer la température à l'aide d'un capteur PT100 ou thermocouple.
3	Ajoutez le module à la description du système (reportez-vous à la section (Positionnement d'un module, Twido, Logiciel de programmation TwidoSuite, Aide en ligne)).
4	A l'aide de l'éditeur de configuration (Editeur de configuration, Twido, Logiciel de programmation TwidoSuite, Aide en ligne), définissez les paramètres des modules d'E/S analogiques que vous avez ajoutés en tant que modules d'expansion lors de la description du système.
5	Dans la colonne **Type**, sélectionnez le type d'entrée correspondant au type de capteur utilisé (thermocouple K, s'il s'agit de ce type de capteur).
6	Dans la colonne **Etendue**, sélectionnez l'unité de mesure du capteur. S'il s'agit de capteurs de température, il est plus facile de sélectionner des degrés **Celsius**, car ainsi le nombre de comptes renvoyé par la carte analogique sera un facteur direct de la vraie mesure.
7	Fournissez une adresse pour le symbole d'entrée de la carte analogique configurée. Cette adresse sera utilisée pour renseigner les champs PID (%IW1.0, pour cet exemple).
8	Faites de même pour une sortie analogique si vous devez utiliser une sortie pour commander le système de contrôle.

Exemple de configuration d'une voie analogique

Plusieurs types de configuration sont possibles en fonction du type de mesure utilisé, comme indiqué ci-dessous :

- Pour l'application de l'exemple utilisé dans ce document, nous avons choisi un **thermocouple de type K** (0 à 200°). La lecture de la mesure sera directement compréhensible (2 000 comptes = 200°, le facteur d'unité étant 0,1).
- Pour d'autres types de mesure, sélectionnez **0-10 V** ou **4-20 mA** dans la colonne **Type**, ou **Personnalisé** dans la colonne **Etendue**. Réglez ensuite l'échelle de valeur (saisissez **0** dans la colonne **Minimum** et **10 000** dans la colonne **Maximum**) pour pouvoir lire directement la mesure (10 V = 10 000 comptes).

L'exemple ci-dessous illustre la configuration d'une voie analogique de thermocouple K :



Utilisé	Adresse	Symbole	Type	Etendue	Minimum	Maximum	Unités
☐	%IW2.0		Non utilisé	Normale	0	1 095	Aucune
☐	%IW2.1		Non utilisé	Normale	0	4 095	Aucune
☐	%QW2.0		Non utilisé	Normale	0	4 095	Aucune

Etape 2 - Conditions préalables à la configuration de l'automate PID

Introduction

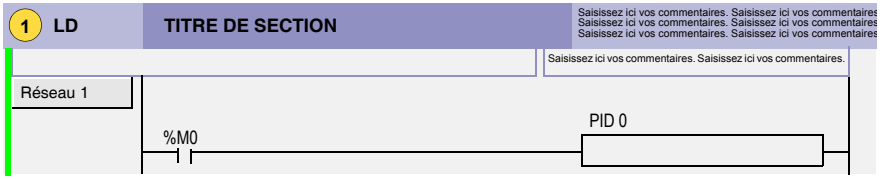
Avant de configurer l'automate PID, assurez-vous de bien avoir exécuté les phases suivantes :

Phase	Description
1	Automate PID activé dans le programme
2	Période de scrutation configurée

Activation de l'automate PID dans le programme.

L'automate PID doit être activé dans le programme à l'aide d'une instruction. Cette instruction peut être permanente ou être une condition d'une entrée ou d'un bit interne. Dans l'exemple suivant, l'automate PID est activé par l'instruction %M0 :

- En langage schéma à contacts :


1
LD
TITRE DE SECTION
Saisissez ici vos commentaires. Saisissez ici vos commentaires
Saisissez ici vos commentaires. Saisissez ici vos commentaires
Saisissez ici vos commentaires. Saisissez ici vos commentaires
Saisissez ici vos commentaires. Saisissez ici vos commentaires.
Réseau 1
%M0
PID 0

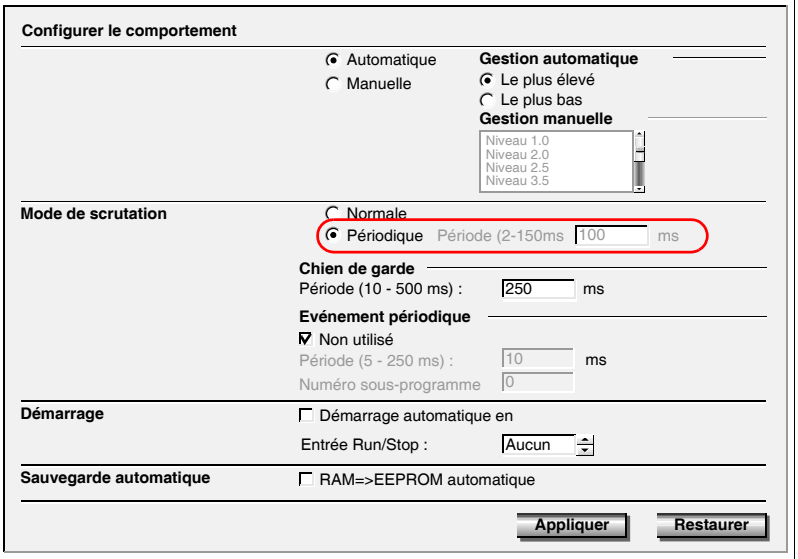
- En liste d'instructions :

```
----
   0 LD    %M0
   1 [ PID 0 ]
```

Note : Vérifiez que vous utilisez la bonne syntaxe :
Vérifiez qu'il y a un espace entre "PID" et le numéro du PID (ex : `PID<espace>0`).

Configuration de la période de scrutation

Lorsque vous utilisez des automates PID, il est fortement recommandé de configurer le mode de scrutation du cycle de l'automate sur Périodique. Le tableau ci-dessous décrit la marche à suivre pour configurer le mode de scrutation.

Etape	Action
1	Utilisez la tâche **Programme** → **Configurer** → **Configurer le comportement** pour paramétrer le **Mode de scrutation** de l'automate.
2	Cochez la case **Périodique**.
3	Réglez le temps de cycle comme indiqué dans l'écran ci-dessous :  **Remarque :** Il est préférable de régler le temps de cycle en fonction de la taille du programme et des performances souhaitées. (Un temps de cycle de 50 ms est un bon compromis.)

Etape 3 - Configuration du PID

Introduction

Pour cet exemple, nous avons choisi de mettre en œuvre la majeure partie des fonctions de l'automate PID dans Twido. Certaines sélections ne sont pas essentielles et peuvent être simplifiées.

Auto-Tuning (AT)

L'automate PID possède une fonction Auto-Tuning qui simplifie le paramétrage de la boucle de régulation (cette fonction est appelée AT dans le reste du document).

Modes de fonctionnement

L'automate PID Twido offre quatre modes de fonctionnement différents, configurables dans l'onglet **Général** de la boîte de dialogue **PID** :

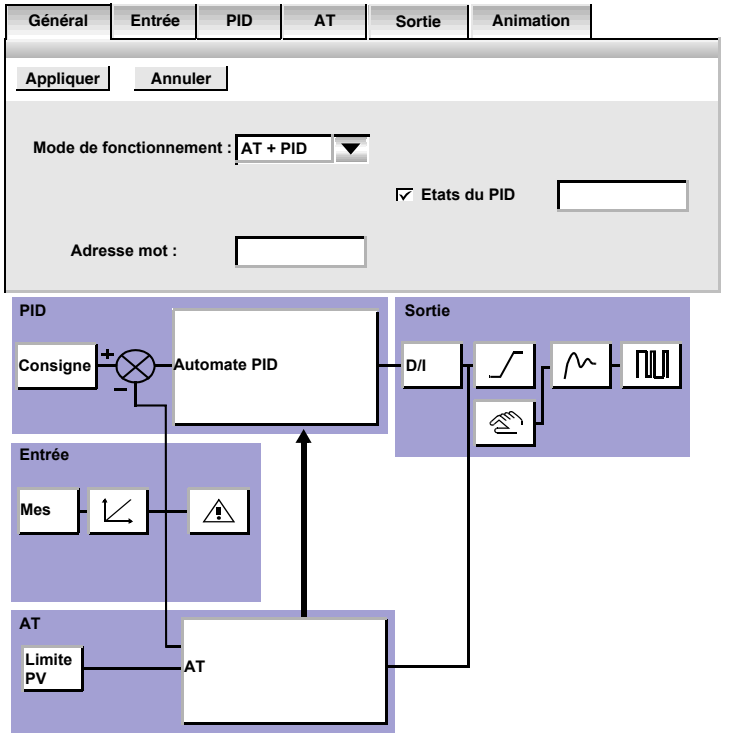
- **PID** = Simple automate PID.
- **AT + PID** = La fonction Auto-Tuning est active lorsque l'automate PID démarre et renseigne automatiquement les valeurs de gain **Kp, Ti, Td** (onglet **PID**) et le type d'action PID (onglet **Sortie**). A la fin de la séquence Auto-Tuning, l'automate passe en mode PID pour la consigne réglée à l'aide des paramètres définis par la fonction AT.
- **AT** = La fonction Auto-Tuning est active lorsque l'automate PID démarre et renseigne automatiquement les valeurs de gain **Kp, Ti, Td** (onglet **PID**) et le type d'action PID (onglet **Sortie**). A la fin de la séquence, l'automate PID s'arrête et attend. Les valeurs de gain **Kp, Ti, Td** (onglet **PID**) et le type d'action PID (onglet **Sortie**) sont renseignés.
- **Adresse mot** = La sélection du mode de fonctionnement de l'automate PID peut être contrôlée par le programme en affectant la valeur souhaitée à l'adresse mot associée à cette sélection.
  - %MWxx=1 : L'automate fonctionne en mode simple automate PID.
  - %MWxx=2 : L'automate fonctionne en mode AT + PID.
  - %MWxx=3 : L'automate fonctionne en mode AT uniquement.
  - %MWxx=4 : L'automate fonctionne en mode PI uniquement.

  Ce type de configuration via l'adresse mot permet à l'utilisateur de gérer le mode de fonctionnement de l'automate PID via le programme d'application, pour un paramétrage mieux adapté aux exigences finales.

Note : La régulation PI peut être sélectionnée dans l'**onglet PID**.

Ouverture de la boîte de dialogue PID

Le tableau ci-dessous représente la boîte de dialogue PID et les procédures à suivre pour accéder aux différents onglets de configuration de l'automate PID :

Etape	Action
1	Sélectionnez la tâche **Programme** → **Configurer** → **Configurer les données** dans l'interface TwidoSuite. **Résultat :** La fenêtre de configuration par défaut du logiciel apparaît.
2	Sélectionnez **Objets avancés** dans la fenêtre Catégorie d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
3	Sélectionnez le PID souhaité (champ **PID n°**) dans le tableau des PID.
4	La boîte de dialogue PID apparaît au premier plan et permet de saisir les différents paramètres de l'automate comme l'indique l'illustration ci-dessous. En mode local, plusieurs onglets s'affichent : **Général, Entrée, PID, AT, Sortie :**  **Important :** Les onglets doivent être paramétrés suivant l'ordre dans lequel ils apparaissent dans la boîte de dialogue : Général en premier, puis Entrée, PID, AT et Sortie : **Remarque :** En mode connecté, deux onglets supplémentaires s'affichent à l'écran, **Animation** et **Trace**, respectivement utilisés pour les diagnostics et l'affichage des opérations de l'automate.

Modification dynamique des paramètres

Pour la modification dynamique des paramètres de l'automate PID (en mode fonctionnement et en mode connecté), il est recommandé de saisir les adresses mémoire dans les champs associés, ce qui évite de devoir passer en mode local pour effectuer des modifications dynamiques de valeurs.

Paramétrage de l'onglet Général

Le tableau suivant décrit comment paramétrer l'onglet **Général** de la boîte de dialogue PID :

Etape	Action
1	Dans l'onglet **Général**, cochez la case **Configuré** pour activer le PID, puis renseignez les champs des autres onglets.
2	Dans la liste déroulante **Mode de fonctionnement**, sélectionnez le type de fonctionnement souhaité (voir *p. 561*). **Dans cet exemple :** Nous allons sélectionner le mode adresse mémoire et saisir le mot %MW17 dans le champ correspondant. Le mode de fonctionnement de l'automate sera alors lié à la valeur du mot %MW17.

Paramétrage de l'onglet Entrée

Le tableau suivant décrit comment paramétrer l'onglet **Entrée** de la boîte de dialogue PID :

Etape	Action
1	Dans l'onglet **Entrée**, saisissez la voie analogique utilisée comme mesure dans le champ correspondant. **Dans cet exemple :** Nous avons choisi **%IW1.0**, car il s'agit d'une mesure de température.
2	Si nécessaire, vous pouvez paramétrer des alarmes pour les seuils inférieur et supérieur de mesure en cochant les cases correspondantes et en renseignant les champs associés. **Remarque :** Les valeurs saisies peuvent être fixes (saisies dans les champs correspondants) ou modifiables (en renseignant les champs associés avec les adresses mémoire : %MWxx).

Paramétrage de l'onglet PID

Le tableau suivant décrit comment paramétrer l'onglet **PID** de la boîte de dialogue PID :

Etape	Action
1	Dans l'onglet **PID**, saisissez la valeur à utiliser pour paramétrer la consigne de l'automate. En règle générale, cette valeur est une adresse mémoire ou la consigne d'une entrée analogique. **Dans cet exemple :** Nous avons saisi %MW0, qui sera utilisé comme un mot de consigne.
2	Le **type de correcteur** peut être sélectionné uniquement si le mode de fonctionnement PID a été préalablement sélectionné dans l'onglet Général. **Dans cet exemple :** Le type de correcteur est réglé sur automatique et désactivé. **Remarque :** Si vous avez précédemment sélectionné le mode de fonctionnement PID, vous pouvez sélectionner le type de correcteur souhaité (**PID** ou **PI**) dans la liste déroulante. Si vous sélectionnez **PI**, le **paramètre Td** est forcé sur zéro et ce champ est désactivé.
3	Définissez les paramètres **Kp, Ti, Td**. **Important :** Si le mode **AT** ou **AT+PID** est sélectionné, les champs Kp, Ti et Td doivent alors contenir des **adresses mémoire**, ce qui permet à la fonction Auto-Tuning de renseigner automatiquement les valeurs trouvées. **Dans cet exemple :** Nous avons saisi %MW10 pour Kp, %MW11 pour Ti et %MW12 pour Td. **Remarque :** En principe, il est plutôt difficile de déterminer les valeurs optimales des paramètres Kp, Ti et Td dans le cas d'une application qui n'a pas encore été développée. Par conséquent, il est fortement recommandé de saisir dans ces champs des adresses de mots mémoire, ce qui vous permettra de saisir les valeurs en mode connecté sans avoir à passer en mode local pour effectuer des modifications dynamiques.
4	Définissez la **Période d'échantillonnage** PID. Cette valeur permet à l'automate d'obtenir les mesures et de mettre à jour les sorties. **Dans cet exemple :** Nous avons paramétré la période d'échantillonnage PID sur 100, soit 1 seconde. Etant donné que le système réglé possède une constante de temps de quelques minutes, la période d'échantillonnage choisie semble correcte. **Important :** Il est recommandé de régler la période d'échantillonnage sur un multiple de la période de scrutation de l'automate et de choisir une valeur concordant avec le système réglé.

Paramétrage de l'onglet AT

Le tableau suivant décrit comment paramétrer l'onglet **AT** de la boîte de dialogue PID :

Etape	Action
1	Dans l'onglet **AT**, cochez la case **Autoriser** si vous souhaitez utiliser la fonction AT.
2	Définissez la **Limite de la mesure**. Cette valeur correspond à la valeur limite que la mesure ne doit pas dépasser pendant l'exécution de la fonction AT.
3	Saisissez la valeur de la **Consigne de sortie** qui correspond à la valeur de sortie de l'automate envoyée pour générer la fonction AT.
Remarque	**Pour plus d'informations sur le paramétrage de ces valeurs, reportez-vous à la section *p. 598*.**
Recommandations	**Il est fortement recommandé de saisir dans ces champs des adresses de mots mémoire, ce qui vous permettra de saisir les valeurs en mode connecté sans avoir à passer en mode local pour effectuer des modifications dynamiques.**

Paramétrage de l'onglet Sortie

Le tableau suivant décrit comment paramétrer l'onglet **Sortie** de la boîte de dialogue PID :

⚠ AVERTISSEMENT

RISQUE DE SURCHARGE DU SYSTEME

Nous vous rappelons que le mode manuel a un effet direct sur la sortie de l'automate. Par conséquent, envoyer une consigne de sortie manuelle (champ **Sortie**) a une influence directe sur le système ouvert contrôlé. Vous devez donc être très attentif aux opérations réalisées lorsque l'automate est en mode manuel.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

⚠ AVERTISSEMENT

FONCTIONNEMENT INATTENDU DU MATERIEL

N'utilisez pas de sortie à relais avec un PID, car le nombre d'opérations de relais autorisé risquerait d'être dépassé et le relais d'être détruit. Suivant le processus contrôlé, les conséquences peuvent être dangereuses.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Etape	Action
1	Dans l'onglet **Sortie**, sélectionnez une action dans la liste déroulante **Action**. Cette sélection dépend du système configuré : ● Action directe : la sortie de l'automate diminue à mesure que la valeur de variation (consigne - mesure) augmente (système froid). ● Action inverse : la sortie de l'automate diminue à mesure que la valeur de variation (consigne - mesure) augmente (système chaud). **Important :** Lorsque la fonction AT est utilisée, l'option **Adresse Bit** est sélectionnée par défaut dans la liste déroulante. Le mode de fonctionnement est déterminé par la fonction AT. Dans le cas présent, il est indiqué dans le bit associé à ce champ.
2	Si nécessaire, saisissez les valeurs de seuil de la sortie de l'automate dans le champ **Alarmes**. Cette fonction peut être utile dans certaines applications pour gérer les alarmes de processus lorsque les valeurs de seuil sont dépassées.
3	Paramétrez le mode de fonctionnement sur **Mode manuel**. La liste déroulante propose plusieurs choix : ● **Inhiber** = mode manuel désactivé. ● **Autoriser** = l'automate fonctionne en mode manuel uniquement. ● **Adresse Bit** = la valeur du bit permet de modifier le fonctionnement du mode manuel (bit à 0 = mode automatique, bit à 1 = mode manuel). **Dans cet exemple :** Nous sélectionnons %M2 pour activer la sélection et %MW18 pour régler la valeur de la consigne manuelle. Utilisez le mode manuel pour effectuer des essais permettant de déterminer la **limite de sortie min./max.** ou la **consigne de sortie AT** la plus précise.
4	Réglez le mot **Sortie TOR**. Ce mot permet à l'automate d'envoyer la consigne de contrôle. Il peut être envoyé directement à une voie de sortie analogique (%QW..) ou à un mot mémoire (%MWxx) pour un traitement supplémentaire. **Important :** Lorsque vous utilisez la fonction PWM, saisissez une adresse mémoire (%MWxx) dans ce champ.
5	Paramétrez la **Sortie PWM** si le système le demande : **1.** Cochez la case **Autoriser** si vous souhaitez contrôler le système via un actionneur PWM. **2.** Saisissez la **Période** de contrôle PWM dans le champ correspondant. **3.** Saisissez la **Sortie** servant à contrôler l'actionneur PWM. Pour cette fonction, il est recommandé d'utiliser les sorties transistor de la base automate (par exemple, %Q0.0 ou %Q0.1 pour la base automate TWDLMDA20DRT).
6	Validez la configuration de l'automate en cliquant sur le bouton **OK** situé dans la partie inférieure gauche de l'écran.
7	Pour configurer plusieurs automates PID, cliquez sur le bouton **Suivant** pour incrémenter le nombre d'automates PID à configurer.

Etape 4 - Initialisation de la configuration du contrôle

Conditions préalables à la configuration

Avant la configuration, procédez comme suit :

Etape	Action
1	Connectez le PC à l'automate et transférez l'application.
2	Faites passer l'automate en mode RUN.

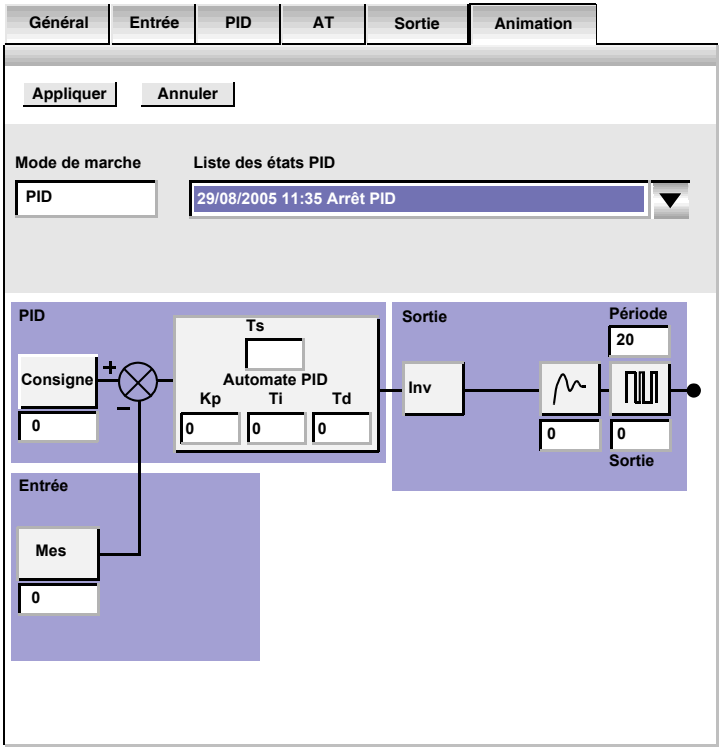
Note : Avant de faire passer l'automate en mode RUN, vérifiez que ses conditions de fonctionnement autorisent ce mode dans le reste de l'application.

Marche à suivre

Procédez comme suit pour initialiser la configuration du contrôle :

Etape	Action
1	Créez une table d'animation contenant les principaux objets nécessaires aux diagnostics. **Dans cet exemple :** ● %MW0 : consigne de l'automate asservi, ● %IW1.0 : mesure ● %M0 : activation de l'automate asservi ● %M1 : type d'action de l'automate asservi (défini par la fonction AT) ● %M2 : sélection du mode Automatique ou Manuel ● %MW10 à %MW12 : coefficients de l'automate asservi PID ● %MW13 : limite de la mesure à ne pas dépasser en mode AT ● %MW14 : consigne de sortie de l'automate asservi en mode AT ● %MW15 : sortie TOR de l'automate PID asservi (renseignée par l'automate) ● %MW16 : paramétrage de la période PWM ● %MW17 : sélection du mode de marche de l'automate PID ● %MW18 : consigne manuelle associée à la sélection du bit %M2

Etape	Action
2	Vérifiez la cohérence de la valeur mesurée dans le champ %IW1.0. **Dans cet exemple :** **1.** On obtient une mesure de 248 incréments lorsque le système est stable et froid. **2.** Cette mesure parait cohérente car nous disposons d'un coefficient de multiplication égal à 10 entre la température et la valeur lue. Il est également possible d'influencer la mesure de manière externe pour s'assurer de la cohérence de la valeur lue (en augmentant la température aux alentours de la sonde pour vérifier que la mesure augmente également). **Remarque :** Ce test est relativement important puisque le fonctionnement de l'automate dépend essentiellement de la précision de la mesure. **3.** Si vous avez le moindre doute sur la précision de la mesure, mettez l'automate en mode STOP et vérifiez le câblage des entrées de la carte analogique (avec un voltmètre ou un ampèremètre pour les entrées 0-10 V / 4-20 mA, un ohmmètre pour les PT100 (100 ohms à 20 °C) ou un thermocouple (une dizaine d'ohms)) : ● Tout d'abord, déconnectez la sonde des borniers de la carte analogique. ● Vérifiez qu'aucun câble n'a été inversé (vérification de la couleur des câbles connectés aux entrées et du câble de compensation pour les PT100). **Avertissement :** Les voies d'entrée IN0 et IN1 partagent un potentiel sur les borniers (-). ● Vérifiez que la carte analogique est alimentée par une tension de 24 V cc sur les deux premiers borniers. ● Vérifiez que les capteurs des entrées 4-20 mA sont alimentés. Les cartes d'entrées analogiques Twido ne sont pas une source de courant.
3	Pour mettre sous tension l'automate asservi, commencez par faire passer l'automate PID en mode **Manuel** pour augmenter les valeurs de limite requises par la fonction AT. Pour régler l'automate sur le mode Manuel : **1.** Faites passer l'automate en mode RUN. **2.** Attribuez les valeurs suivantes aux adresses mémoire dans la table d'animation : ● %M2 : sélection du mode Manuel = 1, (M2=1 => Mode Manuel, M2=0 => Mode Automatique) ● %MW16 : paramétrage de la période PWM = 10 ● %MW17 : sélection du mode de marche de l'automate PID = 1 (PID uniquement) ● %MW18 : consigne manuelle associée à la sélection du bit %M2 = 1000 Cette valeur de consigne peut être sélectionnée plusieurs fois, à condition que le système puisse revenir à son état initial. **Dans cet exemple :** Nous avons sélectionné la valeur 1000, qui correspond à une augmentation moyenne de la température (pour information, 2000 incréments = 200 °C). Lorsque le système est froid, il démarre à une valeur de 250 incréments.
4	Vérifiez que l'automate est en mode RUN. (%M0 : validation de l'automate = 1, à saisir dans la table d'animation.)
5	Sélectionnez **Objets avancés** dans la fenêtre Catégorie d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). Sélectionnez le PID souhaité (champ **PID numéro**) dans le tableau des PID.

Etape	Action
6	Activez l'onglet **Animation** pour les numéros de PID requis et vérifiez que l'animation correspond à l'écran suivant :  **Remarque :** Les écrans de l'automate PID ne sont mis à jour que si l'automate est activé (et que l'API est en mode RUN).
7	Activez l'onglet **Trace** pour les différents PID requis, puis : **1.** Sélectionnez **15 min** dans la liste déroulante du temps écoulé pour afficher une trace de la progression du signal de mesure. **2.** Vérifiez que la valeur de la mesure reste dans la plage des valeurs autorisées par le système. L'augmentation de la mesure peut être vérifiée dans l'onglet Trace. Une fois la mesure stabilisée, lisez la valeur correspondant à la stabilisation du graphique des mesures (par exemple, 350 incréments correspondant à 35 °C ou une augmentation de 10 °C par rapport à l'état initial).
8	Sélectionnez 15 min dans la liste déroulante du temps écoulé pour afficher une trace de la progression du signal de mesure. Vérifiez que la valeur de la mesure reste dans la plage des valeurs autorisées par le système. L'augmentation de la mesure peut être vérifiée dans l'onglet Trace. Une fois la mesure stabilisée, lisez la valeur correspondant à la stabilisation du graphique des mesures (par exemple, 350 incréments correspondant à 35 °C ou une augmentation de 10 °C par rapport à l'état initial).

Etape	Action

9	Si vous vous apercevez que l'actionneur n'est pas contrôlé, vérifiez le circuit de sortie :
	● Dans le cas d'une sortie analogique, vérifiez la tension ou le courant de sortie depuis la carte analogique.
	● Dans le cas d'une sortie PWM, vérifiez :
	  ● que le voyant de la sortie en question est allumé (%Q0.1 dans cet exemple) ;
	  ● le câblage des alimentations et du circuit O V des sorties de la base TWDLMDA20DRT ;
	  ● l'alimentation de l'actionneur.

10	Fermez l'écran de l'automate PID et arrêtez le mode manuel en saisissant les valeurs suivantes dans la table d'animation : ● %M0 : activation de l'automate asservi = 0 (arrêt de l'automate asservi) ● %M2 : sélection du mode Automatique ou Manuel = 0 (arrêt du mode manuel) ● %MW17 : sélection du mode de marche de l'automate PID = 0 ● %MW18 : consigne manuelle associée à la sélection du bit %M2 = 0

Etape 5 - Fonction AT + PID pour l'installation du contrôle

Introduction

Dans cette section, nous étudierons comment configurer l'automate pour qu'il démarre en mode AT+PID. Lorsque ce mode de marche est activé, l'automate est automatiquement réglé en fonction des coefficients Kp, Ti et Td.

Note : Lors de la séquence, le système ne doit pas être soumis à des variations extérieures qui pourraient affecter les derniers réglages. De la même manière, assurez-vous que le système est stable avant de lancer la séquence AT.

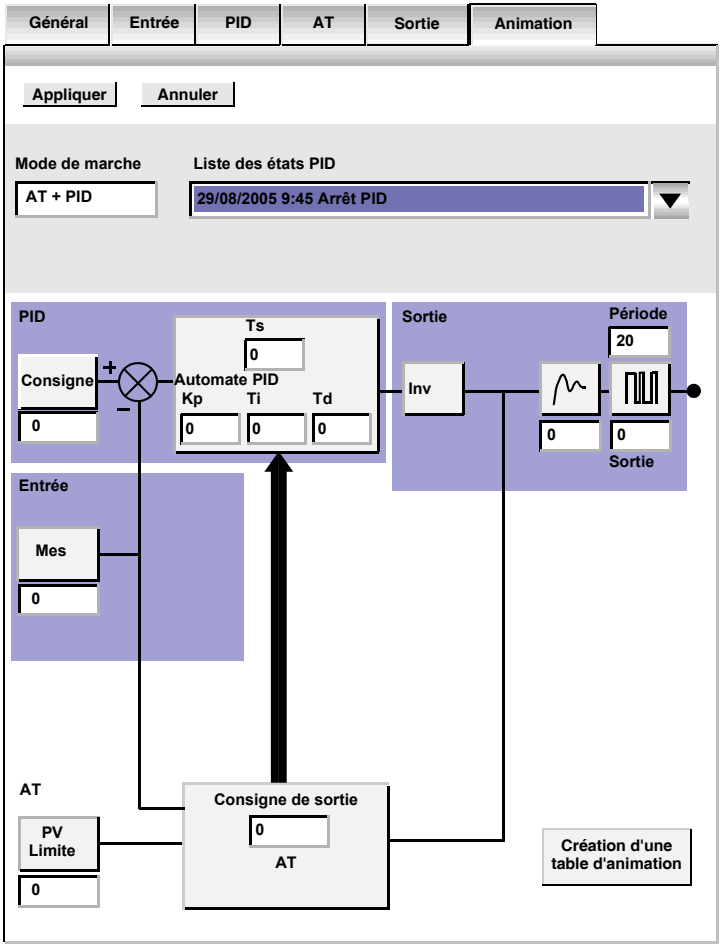
Rappel de la configuration de Kp, Ti et Td

Pour que le système fonctionne en mode AT+PID, les deux conditions suivantes doivent être respectées :

- Les coefficients **Kp, Ti et Td** doivent être configurés en tant qu'**adresses mémoire (%MWxx)**.
- Le type d'**action** sélectionné dans l'onglet **Sortie** doit être défini sur une **adresse bit mémoire (%Mxx)**.

Pour régler l'automate sur le mode AT+PID, procédez comme suit :

Etape	Action
1	Vérifiez que les valeurs suivantes sont bien attribuées aux adresses mémoire dans la table d'animation, ou saisissez-les : ● %M2 : sélection du mode Automatique ou Manuel = 0 ● %MW0 : consigne de l'automate asservi = 600 (dans cet exemple, la consigne est active après la séquence AT et l'automate reste à une température de 60 °C) ● %MW10 à %MW12 : coefficients de l'automate PID (laissez la valeur 0, la séquence AT renseignera ces valeurs) ● %MW13 : limite de la mesure à ne pas dépasser en mode AT, mode = 900 (dans cet exemple, si la température est supérieure à 90 °C, une erreur se produira en mode AT) ● %MW14 : consigne de sortie de l'automate en mode AT, mode = 2000 (test en mode manuel) Il s'agit de la valeur du changement d'étape appliquée au processus. En mode AT, la consigne de sortie est directement appliquée à la sortie de l'automate. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. La valeur doit donc être comprise entre 0 et 10 000. **Remarque :** La consigne de sortie d'auto tuning doit toujours être supérieure à la dernière sortie appliquée au processus. ● %MW15 : sortie TOR de l'automate PID asservi (renseignée par l'automate) ● %MW16 : paramétrage de la période PWM (conservez la valeur précédente, 10) ● %MW17 : sélection du mode de marche de l'automate PID = 2 (AT + PID) ● %MW18 : consigne manuelle associée à la sélection du bit %M2 = 0
2	Configurez l'automate Twido de façon à ce qu'il effectue une scrutation en mode **Périodique.**
3	Définissez la **période de scrutation** de l'automate Twido de façon à ce que la valeur de la **période d'échantillonnage (Ts)** de l'automate PID soit un multiple exact. **Remarque :** Pour plus de détails sur la manière de déterminer la période d'échantillonnage, consultez les sections *p. 615* et *p. 616*.
4	Vérifiez que l'automate est en mode RUN.
5	Saisissez le bit mémoire %M0. %M0 : validation de l'automate = 1 dans la table d'animation
6	Sélectionnez **Objets avancés** dans la fenêtre Catégorie d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). Sélectionnez le PID souhaité (champ **PID numéro**) dans le tableau des PID.

Etape	Action
7	Activez l'onglet **Animation** pour les numéros de PID requis et vérifiez que l'animation correspond à l'écran suivant :  **Remarque :** Les écrans de l'automate PID ne sont mis à jour que si l'automate est activé (et que l'API est en mode RUN).
8	Cliquez sur le bouton **Trace** et attendez que le système démarre la fonction AT. **Trace** **Remarque :** 10 à 20 minutes peuvent s'écouler avant que la procédure AT ne change.

Stockage des coefficients calculés Kp, Ti et Td

Une fois la séquence Auto-Tuning terminée, les valeurs calculées sont affectées aux mots mémoire des coefficients Kp, Ti et Td. Ces valeurs sont écrites dans la mémoire RAM et enregistrées dans l'automate tant que l'application reste valide (mise hors tension d'une durée inférieure à 30 jours) et qu'aucun démarrage à froid n'est effectué (%S0).

Note : Si le système n'est pas soumis à des variations extérieures, les valeurs peuvent être écrites **en dur** dans les paramètres de l'automate PID et l'automate peut basculer en mode PID uniquement.

Répétition de AT

La séquence Auto-Tuning est répétée chaque fois que l'automate passe en mode RUN ou à chaque démarrage à froid (%S0).

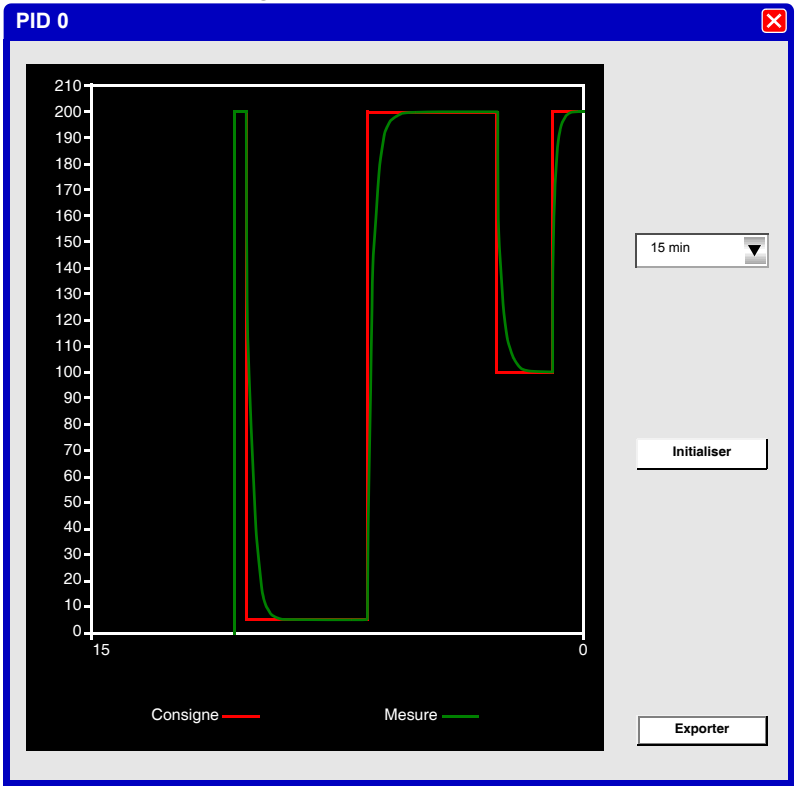
Par conséquent, vous devez tester les mots de diagnostic en utilisant le programme indiquant la procédure à suivre en cas de redémarrage.

Etape 6 - Mise au point des réglages

Accès à la table d'animation

Pour faciliter la mise au point du système, la table d'animation est accessible à tout moment lorsque les écrans de l'automate PID sont au premier plan.

Note :
Lorsque vous affichez uniquement le graphique de la consigne et de la mesure à l'aide du bouton **Détacher** dans l'onglet **Trace** (voir la fenêtre de l'onglet Trace ci-dessous), la table d'animation est accessible via la tâche **Programme** → **Mise au point** → **Animer le programme**.



Les données d'affichage peuvent être exportées au format Excel en cliquant sur le bouton **Exporter**. Une boîte de dialogue apparaît, dans laquelle vous pouvez spécifier le nom et l'emplacement d'un fichier *.cvs*. Dans cette boîte de dialogue, cliquez sur **Enregistrer** pour exporter les données ou sur **Annuler** pour annuler l'export.

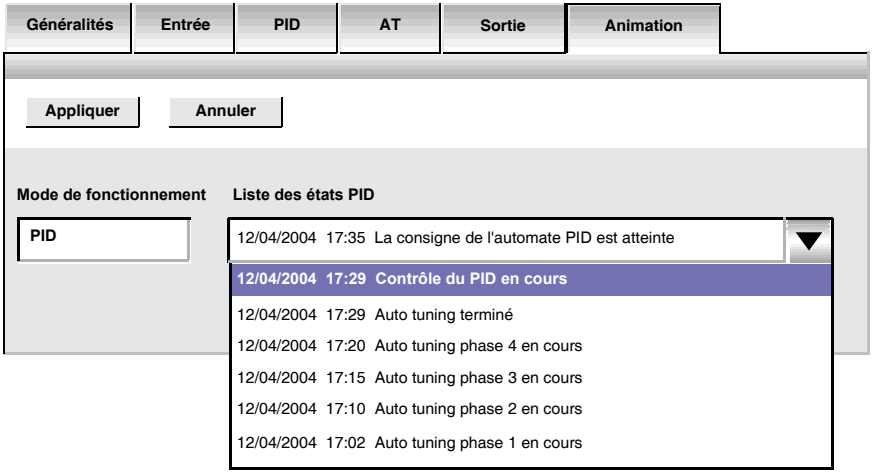
Retour aux écrans de l'automate PID

Pour retourner aux écrans de l'automate PID sans perdre l'historique de tracé du graphique, procédez comme suit :

Etape	Action
1	Sélectionnez **Objets avancés** dans la fenêtre Catégories d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne). Sélectionnez le PID souhaité (champ **PID n°**) dans le tableau des PID.
2	Cliquez sur l'onglet **Animation**.

Historique des états du PID

Dans l'onglet **Animation** des automates PID, vous pouvez accéder aux 15 derniers états de l'automate courant en effectuant votre sélection dans la liste déroulante illustrée ci-dessous :



Note : Les états de l'automate PID sont enregistrés lorsque le PC et TwidoSuite sont en mode connecté.

18.4 Fonction PID

Présentation

Objet de cette section

Cette section décrit le comportement, les fonctionnalités et la mise en œuvre de la fonction PID.

Note : Pour obtenir des informations de configuration sur l'automate PID, ainsi que sur l'auto tuning PID, consultez le *p. 554*.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble	579
Principe de la boucle de régulation	580
Méthodologie de développement d'une application de régulation	581
Compatibilités et performances	582
Caractéristiques détaillées de la fonction PID	583
Comment accéder à la configuration du PID	586
Eléments PID de la fonction PID	587
Onglet Général de la fonction PID	591
Onglet Entrée de la fonction PID	594
Onglet PID de la fonction PID	596
Onglet AT de la fonction PID	598
Onglet Sortie de la fonction PID	603
Comment accéder à la mise au point du PID	606
Onglet Animation de la fonction PID	607
Ecran Trace de la fonction PID	609
Etats du PID et codes d'erreurs	611
Réglage PID avec la fonction d'auto tuning (AT)	615
Méthode de réglage du paramètre PID	623
Rôle et influence des paramètres PID	626
Annexe 1 : Notions fondamentales de la théorie PID	630
Annexe 2 : Premier ordre avec modèle de temporisation	632

Vue d'ensemble

Généralités

La fonction de régulation PID est une fonction du langage de programmation TwidoSuite.

Elle permet de programmer des boucles de régulation PID sur des automates compatibles avec TwidoSuite version 2.0 ou supérieure.

Cette fonction est particulièrement adaptée pour :

- répondre aux besoins de process séquentiels nécessitant des fonctions de régulation auxiliaire (exemples : machines d'emballage à film plastique, machines de traitement de surface, presses...),
- répondre aux besoins des process de régulation simple (exemples : fours de traitements de métaux, fours à céramiques, petits groupes frigorifiques...),

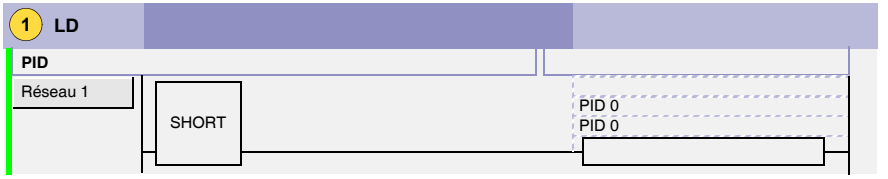
Sa mise en œuvre est très simple car elle s'effectue par des écrans de :

- configuration,
- et de mise au point,

associés à une ligne de programme (bloc opération en langage à contact ou simple appel de PID en liste d'instruction) qui indique le numéro du PID utilisé.

La syntaxe correcte d'une instruction PID est la suivante : **PID<espace>n**, où n est le numéro du PID.

Exemple de ligne de programme en langage schéma à contacts :



Note : dans une même application d'automatisme Twido, le nombre maximum de fonctions PID configurables est de 14.

Principales fonctionnalités

Les principales fonctionnalités sont les suivantes :

- entrée analogique,
- conversion linéaire de la mesure configurable,
- alarme haute et basse en entrée configurable,
- sortie analogique ou PWM,
- écrêtage de la sortie configurable,
- action directe ou inverse configurable.

Principe de la boucle de régulation

Présentation

Le fonctionnement d'une boucle de régulation comprend trois phases distinctes :

- l'acquisition des données :
  - mesure(s) provenant des capteurs du process (analogiques, codeurs),
  - consigne(s) provenant généralement de variables internes de l'automate ou de données issues d'une table d'animation TwidoSuite.
- l'exécution de l'algorithme de régulation PID,
- l'envoi des commandes adaptées aux caractéristiques des actionneurs à piloter via des sorties TOR (PWM) ou analogiques.

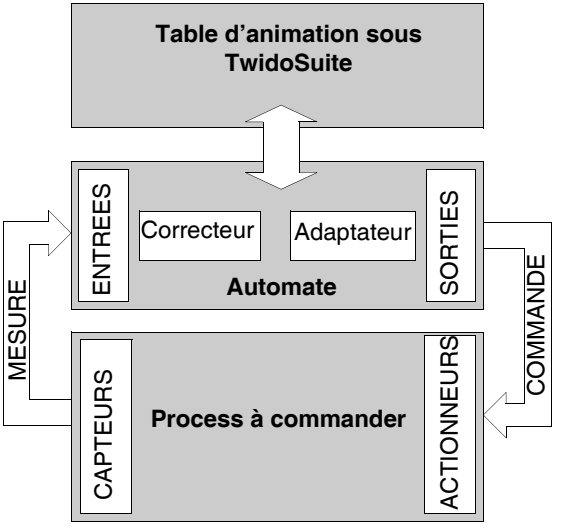
L'algorithme PID élabore le signal de commande à partir :

- de la mesure échantillonnée par le module d'entrée,
- de la valeur de la consigne fixée soit par l'opérateur, soit par programme,
- des valeurs des différents paramètres du correcteur.

Le signal issu du correcteur est soit traité directement par une carte de sortie analogique de l'automate raccordé à l'actionneur, soit traité via une adaptation PWM sur une sortie TOR de l'automate.

Illustration

L'illustration ci-dessous schématise le principe d'une boucle de régulation.

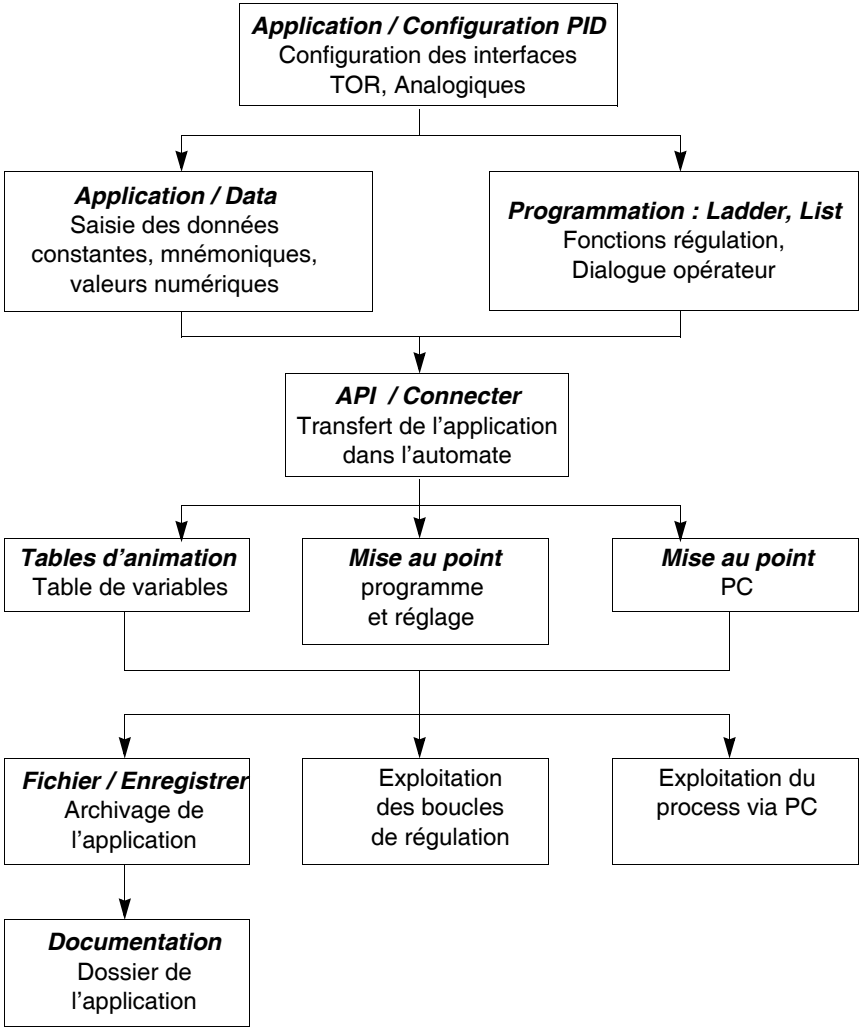


Méthodologie de développement d'une application de régulation

Schéma de principe

Le schéma ci-dessous présente l'ensemble des tâches à effectuer lors de la création et la mise au point d'une application de régulation.

Remarque : L'ordre défini dépend de votre propre méthode de travail, il est donné à titre indicatif.



Compatibilités et performances

Présentation

La fonction PID du Twido est une fonctionnalité disponible pour les automates compatibles avec TwidoSuite version 2.0 minimum, c'est pourquoi sa mise en œuvre est sujette à un certain nombre de compatibilités matérielles et logicielles décrites dans les paragraphes suivants.

D'autre part cette fonctionnalité nécessite des ressources qui sont présentées dans le paragraphe **Performances**.

Compatibilités

La fonction PID du Twido est disponible sur les automates Twido de version logicielle supérieure ou égale à 2.0.

Si vous disposez de Twido de version logicielle inférieure, vous pouvez mettre à jour le firmware afin de pouvoir utiliser cette fonction PID.

Note : Les modules d'entrées et de sorties analogiques de version 1.0 sont utilisables en entrées ou sorties de PID sans nécessiter de mise à jour.

Pour pouvoir configurer et programmer un PID sur ces différentes versions de matériel vous devez posséder la version **1.2 du logiciel TwidoSuite**.

Performances

Les boucles de régulation PID possèdent les performances suivantes :

Description	Durée
Temps d'exécution d'une boucle	0,4 ms

Caractéristiques détaillées de la fonction PID

Généralités

La fonction PID réalise une correction PID à partir d'une mesure et d'une consigne analogiques au format par défaut [0 - 10 000] et fournit une commande analogique au même format ou une modulation de largeur (PWM) sur une sortie TOR.

Tous les paramètres PID sont décrits dans les fenêtres qui permettent de les configurer. Nous nous contentons ici de faire une synthèse des fonctions disponibles, d'indiquer leurs mesures et de décrire leur intégration à la fonction PID dans un synoptique de fonctionnement.

Note : Pour une utilisation en pleine échelle (résolution optimale), vous pouvez configurer votre entrée analogique connectée à la branche mesure du PID en 0-10 000. Toutefois, l'automate fonctionne correctement si vous utilisez la configuration par défaut (0-4 095).

Note : Pour que la régulation puisse fonctionner correctement, l'automate Twido doit impérativement **être en mode périodique**. La fonction PID est alors exécutée périodiquement à chaque cycle et l'échantillonnage des données d'entrée PID respecte la période définie dans la configuration (voir tableau suivant).

Détails des fonctions disponibles

Le tableau suivant indique les différentes fonctions disponibles et les échelles correspondantes :

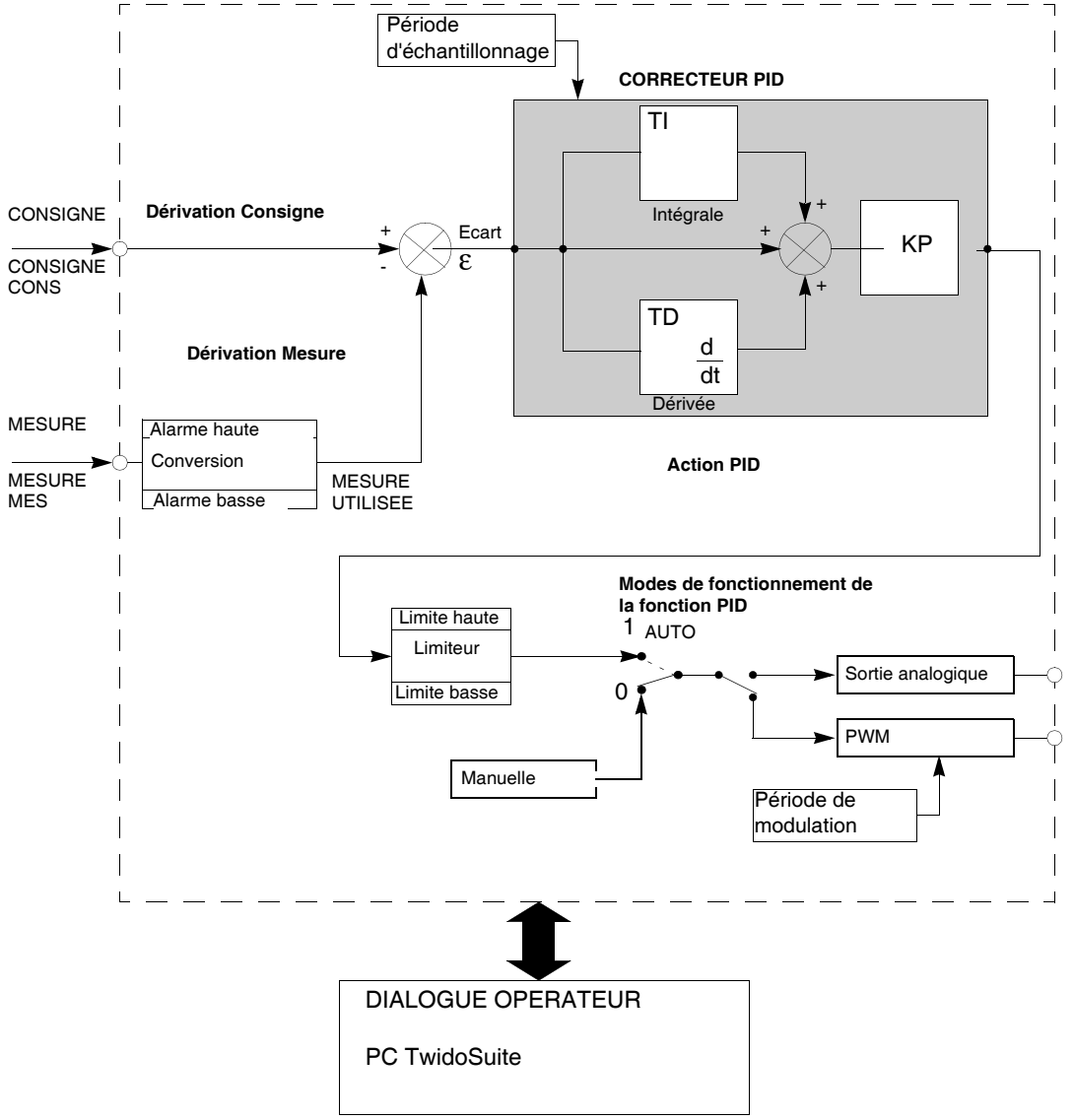
Fonction	Echelle et commentaire
Conversion linéaire de l'entrée	Cette opération permet de convertir une valeur respectant le format 0 à 10 000 (résolution module d'entrée analogique) à une valeur comprise entre -32 768 et 32 767.
Gain proportionnel	En utilisant un facteur de 100, on obtient une valeur comprise entre 1 et 10 000 qui correspond à une valeur de gain comprise entre 0,01 et 100. **Remarque :** Si vous saisissez une valeur de gain incorrecte (gain négatif ou nul), TwidoSuite ignore ce réglage utilisateur et affecte automatiquement la valeur par défaut de 100 à ce facteur.
Temps d'intégrale	En utilisant une base temps de 0,1 seconde, on obtient une valeur comprise entre 0 et 20 000 qui correspond à un temps d'intégrale compris entre 0 et 2 000,0 secondes.
Temps de dérivée	En utilisant une base temps de 0,1 seconde, on obtient une valeur comprise entre 0 et 10 000 qui correspond à un temps de dérivée compris entre 0 et 1 000,0 secondes.
Période d'échantillonnage	En utilisant une base temps de 0,01 seconde, on obtient une valeur comprise entre 1 et 10 000 qui correspond à une période d'échantillonnage comprise entre 0,01 et 100 secondes.
Sortie PWM	En utilisant une base temps de 0,1 seconde, on obtient une valeur comprise entre 1 et 500 qui correspond à une période de modulation comprise entre 0,1 et 50 secondes.

Fonction	Echelle et commentaire
Sortie analogique	Valeur comprise entre 0 et +10 000
Alarme haute sur la mesure	Cette alarme est définie après la conversion. Elle est définie sur une valeur comprise entre -32 768 et 32 767 si la conversion est activée, et entre 0 et 10 000 dans le cas contraire.
Alarme basse sur la mesure	Cette alarme est définie après la conversion. Elle est définie sur une valeur comprise entre -32 768 et 32 767 si la conversion est activée, et entre 0 et 10 000 dans le cas contraire.
Limite haute sur la sortie	Cette limite est comprise entre 0 et 10 000 pour une sortie analogique. Lorsque la fonction PWM est activée, la limite correspond à un pourcentage de la période modulée : 0 % pour 0 et 100 % pour 10 000.
Limite basse sur la sortie	Cette limite est comprise entre 0 et 10 000 pour une sortie analogique. Lorsque la fonction PWM est activée, la limite correspond à un pourcentage de la période modulée : 0 % pour 0 et 100 % pour 10 000.
Mode manuel	Lorsque le mode manuel est activé, la sortie est égale à une valeur fixe paramétrée par l'utilisateur. Cette **sortie** est comprise entre 0 et 10 000 (0 à 100 % pour une sortie PWM). Utilisez le mode manuel pour effectuer des essais permettant de déterminer la **limite de sortie min./max.** ou la **consigne de sortie AT** la plus précise.
Action directe ou inverse	L'action directe ou inverse est disponible et agit directement sur la sortie.
Auto tuning (AT)	Cette fonction permet de régler automatiquement les paramètres Kp, Ti, Td et de l'action directe/inverse afin d'obtenir une convergence optimale de la régulation.

Note : Pour une meilleure compréhension de l'action de chacune des fonctions décrites dans le tableau précédent, reportez-vous au synoptique qui suit.

Principes de fonctionnement

Le schéma suivant présente le principe de fonctionnement de la fonction PID.



Remarque : La description des paramètres utilisés est présentée dans le tableau de la page précédente et dans les écrans de configuration.

Comment accéder à la configuration du PID

Présentation

L'accès aux écrans de configuration d'un PID sur automates Twido est décrit dans les paragraphes qui suivent.

Marche à suivre

Le tableau suivant présente la marche à suivre pour accéder aux écrans de configuration d'un PID :

Etape	Action
1	Vérifiez que vous êtes en mode **local**.
2	Sélectionnez **Objets avancés** dans la fenêtre Catégorie d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
3	Sélectionnez le PID souhaité (champ **PID numéro**) dans le tableau des PID. (voir *p. 588*) **Résultat :** la fenêtre de paramétrage des PID s'ouvre, elle est par défaut positionnée sur l'onglet **Général** (voir *p. 591*).

Eléments PID de la fonction PID

Présentation

La fenêtre de configuration PID vous permet d'effectuer les opérations suivantes :
- configurer chaque PID de l'automate Twido (en mode local) ;
- mettre au point chaque PID de l'automate Twido (en mode connecté).

Cette section décrit les éléments PID, à savoir :
- l'accès à l'écran de configuration PID ;
- le tableau de sélection de la fonction PID ;
- les onglets de la fonction PID ;
- Trace PID.

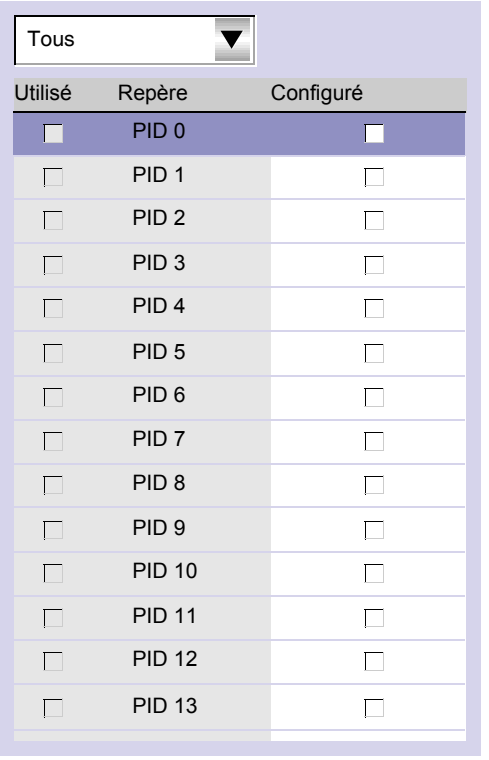
Comment accéder à l'écran de configuration PID

Pour accéder à la fenêtre de configuration PID :

Si ...	Alors ...	Résultat
Vous êtes en mode connecté.	Sélectionnez **Programme** → **Mise au point** → **Surveiller la configuration du logiciel** → **Objets avancés** → **PID**.	Vous accédez à l'onglet Animation et aux paramètres de mise au point et de réglage.
Vous êtes en mode local.	Sélectionnez **Programme** → **Configurer** → **Configurer les données** → **Objets avancés** → **PID**.	Vous accédez à l'onglet par défaut Général et aux paramètres de configuration.

Tableau de sélection PID de la fonction PID

Le tableau ci-dessous permet de sélectionner le PID souhaité pour la configuration ou la mise au point.

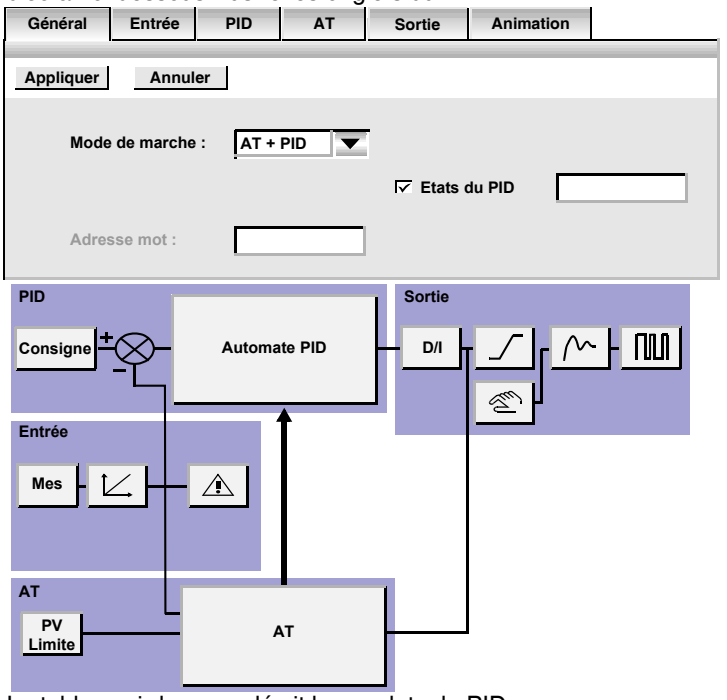


Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
Repère	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Configuré	Cette case doit être cochée pour pouvoir configurer la fonction PID. Si tel n'est pas le cas, aucune action ne peut être effectuée dans ces écrans et la fonction PID, bien qu'existant dans l'application, ne peut pas être utilisée.
Utilisé	Lecture seule. Cette case est cochée lorsque le PID de numéro correspondant est utilisé par le programme d'application.
Boîte de dialogue de tri des options	Sélectionnez l'option de tri appropriée selon que vous souhaitez afficher **tous** les PID, ou uniquement les PID **utilisés**, voire ceux **inutilisés** dans le tableau de sélection PID.
Remarque :	**Vous devez tout d'abord terminer la configuration PID courante avant de passer à un autre PID ou avant d'exécuter toute autre tâche logicielle.**

Onglets de la fonction PID

Les onglets PID vous permettent de configurer les paramètres PID. La capture d'écran ci-dessous illustre les onglets du PID.



Le tableau ci-dessous décrit les onglets du PID.

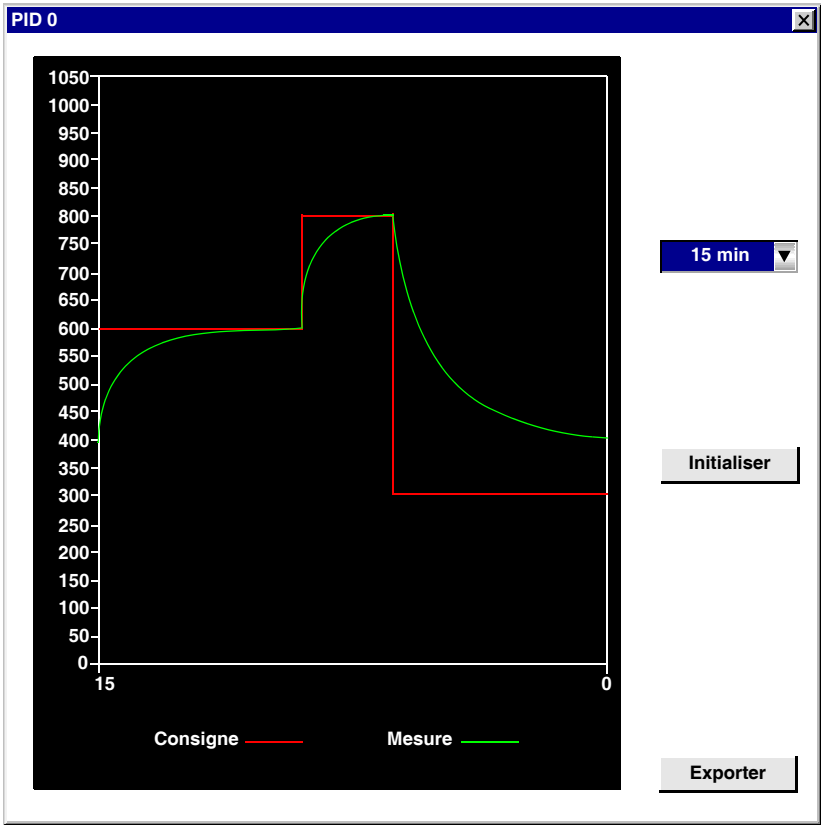
Champ	Description
Onglet Général	Cet onglet permet de spécifier les paramètres généraux du PID. (Reportez-vous à la section *p. 591*)
Onglet Entrée	Cet onglet permet de spécifier les paramètres d'entrée du PID. (Reportez-vous à la section *p. 594*)
Onglet PID	Cet onglet permet de spécifier les paramètres internes du PID. (Reportez-vous à la section *p. 596*)
Onglet AT	Cet onglet permet de spécifier les paramètres d'auto tuning du PID. (Reportez-vous à la section *p. 598*)
Onglet Sortie	Cet onglet permet de spécifier les paramètres de sortie du PID. (Reportez-vous à la section *p. 603*)
Onglet Animation	Cet onglet permet d'afficher et de mettre au point le PID. (Reportez-vous à la section *p. 607*)

Note : Dans certains cas, il est possible que certains onglets et champs ne soient pas accessibles pour l'une des deux raisons suivantes :

- Le mode de marche (local ou connecté) qui est actuellement activé ne permet pas d'accéder à ces paramètres.
- Le mode "PID uniquement" est sélectionné, interdisant ainsi l'accès aux paramètres de l'onglet Auto tuning qui ne sont plus nécessaires.

Bouton Trace de la fonction PID

Le bouton Trace PID Trace vous permet d'afficher le contrôle PID.



Cet onglet permet de visualiser le fonctionnement du PID et d'effectuer ses réglages (reportez-vous à la section *p. 609*).

Onglet Général de la fonction PID

Présentation

Sélectionnez **Objets avancés** dans la fenêtre Catégorie d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et sélectionnez **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).

Sélectionnez le PID souhaité (champ **PID numéro**) dans le tableau des PID.

La fenêtre de configuration de l'automate PID vous permet d'effectuer les opérations suivantes :

- configurer chaque PID de l'automate Twido (en mode connecté) ;
- mettre au point chaque PID de l'automate Twido (en mode local) ;

Lorsque vous affichez cet écran et que vous êtes :

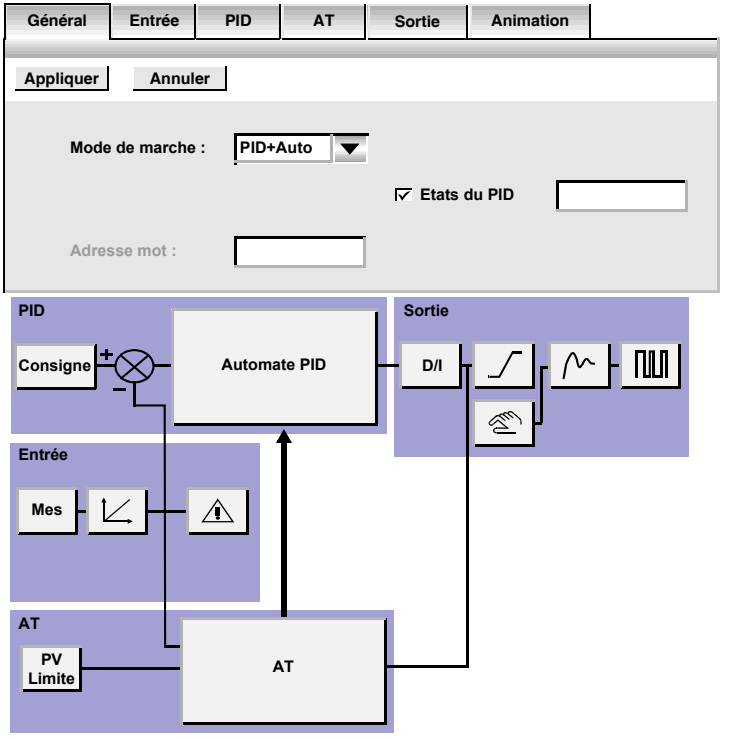
- en mode local : l'onglet **Général** est affiché par défaut et vous donne accès aux paramètres de configuration ;
- en mode connecté : l'onglet **Animation** est affiché et vous donne accès aux paramètres de mise au point et de réglage.

Note : Dans certains cas, il est possible que certains onglets et champs ne soient pas accessibles pour l'une des raisons suivantes :
- Le mode de marche (local ou connecté) qui est actuellement activé ne permet pas d'accéder à ces paramètres.
- Le mode "PID uniquement" est sélectionné, interdisant ainsi l'accès aux paramètres de l'onglet Auto tuning qui ne sont plus nécessaires.

Les paragraphes qui suivent décrivent l'onglet **Général**.

Onglet Général de la fonction PID

L'écran suivant permet de renseigner les paramètres généraux du PID.



Description Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
PID numéro	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Configuré	Cette case doit être cochée pour pouvoir configurer la fonction PID. Si tel n'est pas le cas, aucune action ne peut être effectuée dans ces écrans et la fonction PID, bien qu'existant dans l'application, ne peut pas être utilisée.
Mode de marche	Indiquez ici le mode de marche désiré. Vous pouvez choisir entre trois modes de marche et une adresse de mot, comme suit : ● PID ● AT ● PID+Auto tuning ● Adresse Mot
Adresse Mot	Vous pouvez définir un mot interne (%MW0 à %MW2999) dans la zone de texte. Ce mot est utilisé pour définir le mode de marche par programme. Le mot interne accepte quatre valeurs possibles selon le mode de marche que vous souhaitez définir : ● %MWx = 1 (pour définir PID uniquement) ● %MWx = 2 (pour définir PID+Auto tuning) ● %MWx = 3 (pour définir Auto tuning uniquement) ● %MWx = 4 (pour définir PI uniquement)
Etats du PID	Si vous cochez cette option, vous pouvez définir un mot mémoire dans cette zone de texte (%MW0 à %MW2999). Ce mot est utilisé par l'automate PID pour enregistrer l'état PID courant lors de l'exécution de l'automate PID et/ou la fonction d'auto tuning (pour plus d'informations, reportez-vous à la section *p. 611*).
Schéma	Le schéma vous permet de visualiser les différentes configurations possibles du PID.

Onglet Entrée de la fonction PID

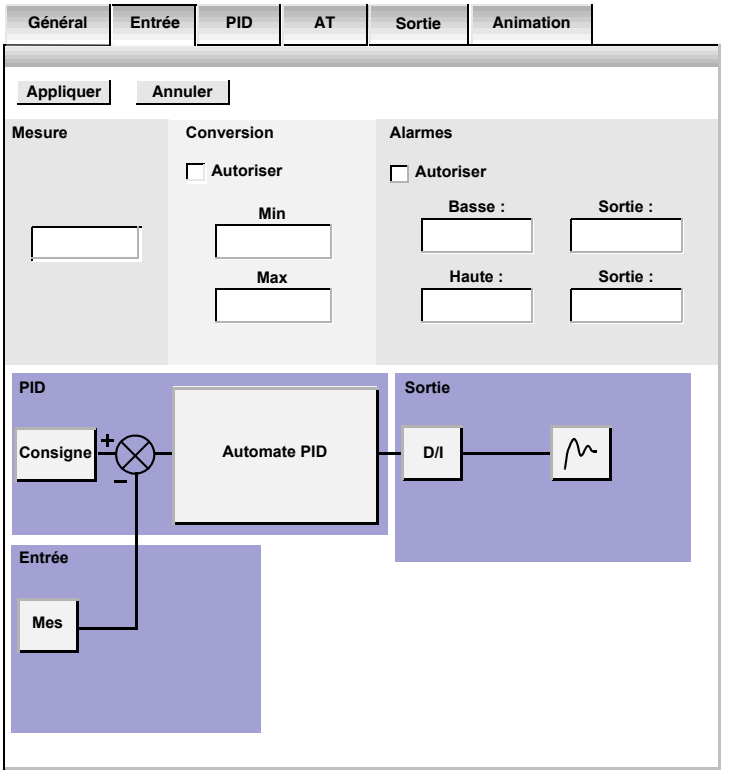
Présentation

Cet onglet permet de renseigner les paramètres d'entrée de la fonction PID.

Note : Il est accessible en mode local.

Onglet Entrée de la fonction PID

L'écran suivant permet de renseigner les paramètres d'entrée du PID.



Description Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
Numéro PID	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Mesure	Indiquez ici la variable qui contiendra la valeur de la mesure du processus à réguler. L'échelle par défaut est comprise entre 0 et 10 000. Vous pouvez saisir soit un mot interne (%MW0 à %MW2999) ou une entrée analogique (%IWx.0 à %IWx.1).
Conversion	Cochez cette case si vous désirez effectuer une conversion de la variable du processus indiquée en entrée du PID. Si cette case est cochée, les deux champs **Valeur min** et **Valeur max** sont accessibles. La conversion est linéaire et convertit une valeur comprise entre 0 et 10 000 en une valeur dont les minimum et maximum peuvent être compris entre -32 768 et +32 767.
Valeur min **Valeur max**	Indiquez les valeurs minimum et valeurs maximum de l'échelle de conversion. La variable du processus est ensuite réévaluée automatiquement dans l'intervalle **[Valeur min à valeur max]**. **Remarque** : La **Valeur min** doit obligatoirement être inférieure à la **Valeur max**. **Valeur min** et **Valeur max** peuvent être soit des mots internes (%MW0 à %MW2999), soit des constantes internes (%KW0 à %KW255), soit encore une valeur comprise entre -32 768 et +32 767.
Alarmes	Cochez cette case si vous désirez activer des alarmes sur des variables d'entrée. **Remarque** : Les valeurs d'alarme sont à déterminer par rapport à la variable obtenue après la phase de conversion. Elles doivent par conséquent être comprises entre **Valeur min** et **Valeur max** lorsque la conversion est activée, sinon elles seront comprises entre **0 et 10 000.**
Basse **Sortie**	Indiquez la valeur d'alarme haute dans le champ **Basse**. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. **Sortie** doit contenir l'adresse du bit qui sera mis à 1 lorsque la limite basse est atteinte. **Sortie** peut être soit un bit interne (%M0 à %M255), soit une sortie (%Qx.0 à %Qx.32).
Haute **Sortie**	Indiquez la valeur d'alarme basse dans le champ **Haute**. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. **Sortie** doit contenir l'adresse du bit qui sera mis à 1 lorsque la limite haute est atteinte. **Sortie** peut être soit un bit interne (%M0 à %M255), soit une sortie (%Qx.0 à %Qx.32).
Schéma	Le schéma vous permet de visualiser les différentes configurations possibles de la fonction PID.

Onglet PID de la fonction PID

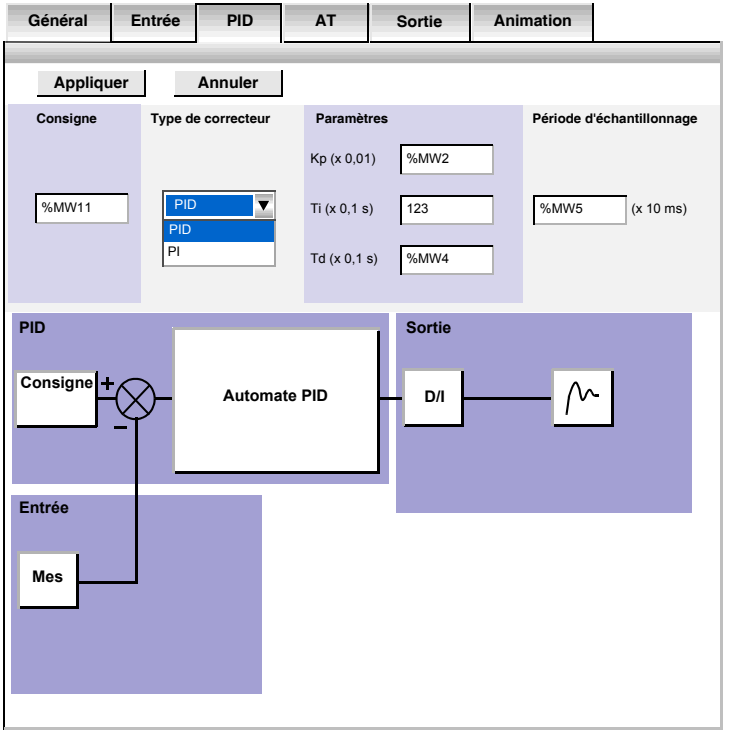
Présentation

Cet onglet permet de renseigner les paramètres internes de la fonction PID.

Note : Il est accessible en mode local.

Onglet PID de la fonction PID

L'écran suivant permet de renseigner les paramètres internes de la fonction PID.



Description Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
Numéro PID	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Consigne	Saisissez ici la valeur de consigne de la fonction PID. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit donc être comprise entre 0 et 10 000 lorsque la conversion est inhibée. Sinon, la valeur doit être comprise entre **Valeur min** et **Valeur max** pour la conversion.
Type de correcteur	Si vous avez préalablement sélectionné le mode de fonctionnement PID dans l'onglet Général, vous pouvez sélectionner le type de correcteur souhaité (PID ou PI) dans la liste déroulante. Si vous avez sélectionné d'autres modes de fonctionnement, le type de correcteur est réglé automatiquement et ne peut être modifié manuellement. Si vous sélectionnez **PI** dans la liste déroulante, le paramètre Td est forcé sur zéro et ce champ est désactivé.
Kp * 100	Indiquez ici le coefficient proportionnel de la fonction PID multiplié par 100. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. La plage valide pour la paramètre Kp est : 0 < Kp < 10 000. **Remarque :** Si Kp est défini de manière incorrecte sur 0 (Kp ≤ 0 est non valide), la valeur par défaut Kp=100 est automatiquement affectée par la fonction PID.
Ti (0,1 s)	Indiquez ici le coefficient d'action intégrale par rapport à une base temps de 0,1 seconde. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 0 et 20 000. **Remarque :** Pour désactiver l'action intégrale de la fonction PID, réglez ce coefficient sur 0.
Td (0,1 s)	Indiquez ici le coefficient d'action dérivée par rapport à une base temps de 0,1 seconde. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 0 et 10 000. **Remarque :** Pour désactiver l'action dérivée de la fonction PID, réglez ce coefficient sur 0.
Période d'échantillonnage	Indiquez ici la période d'échantillonnage PID par rapport à une base de temps de 10^{-2} secondes (10 ms). Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 1 (0,01 s) et 10 000 (100 s).
Schéma	Le schéma vous permet de visualiser les différentes configurations possibles de la fonction PID.

Note : Lorsque la fonction d'auto tuning est activée, les paramètres Kp, Ti et Td ne sont plus définis par l'utilisateur, car ils sont définis automatiquement et par programme par l'algorithme d'auto tuning. Dans ce cas, vous devez saisir dans ces champs un **mot interne** uniquement (%MW0 à %MW2999).
Attention : Ne saisissez pas de constante interne ou de valeur directe lorsque la fonction d'auto tuning est activée, car cela déclencherait une erreur lors de l'exécution de la fonction PID.

Onglet AT de la fonction PID

Présentation

La configuration des paramètres de la fonction PID peut se révéler une tâche ardue, laborieuse et propice aux erreurs. Toutes ces raisons peuvent rendre la régulation difficile à configurer pour les personnes moins expérimentées, mais pas nécessairement expertes en matière de régulation. Il est ainsi parfois difficile d'effectuer un réglage parfait.

L'algorithme d'auto tuning de la fonction PID permet de déterminer automatiquement et correctement les valeurs des quatre éléments du PID suivants :

- le facteur de gain ;
- la valeur de l'intégrale ;
- la valeur de la dérivée ;
- l'action directe ou inverse.

La fonction d'auto tuning permet ainsi de régler rapidement et efficacement la boucle du processus.

Exigences relatives à la fonction d'auto tuning

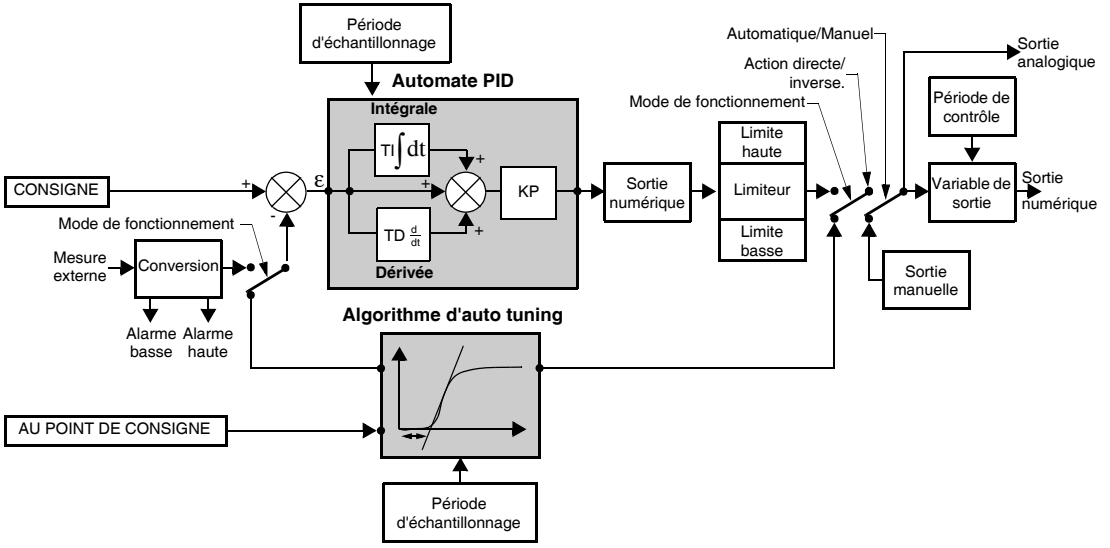
L'auto tuning de la fonction PID est particulièrement adapté à la régulation de température.

D'une manière générale, les processus régulés par la fonction d'auto tuning doivent répondre aux exigences suivantes :

- la régulation doit être principalement linéaire sur toute la plage de fonctionnement ;
- la réponse de la régulation à une modification de niveau de sortie analogique doit suivre un schéma transitoire asymptotique ;
- très peu de perturbations doivent se produire au sein des variables de régulation. (Dans le cas d'une régulation de température, vous ne devez pas constater de taux anormalement élevés d'échange de température entre la régulation et son environnement.)

Principe de fonctionnement de l'auto tuning

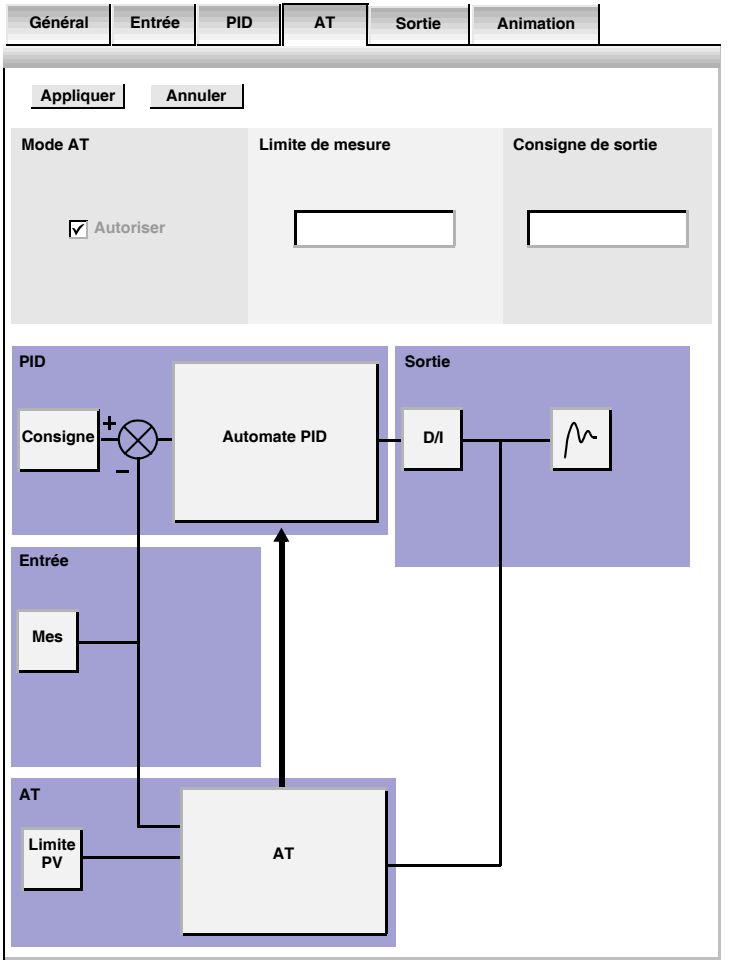
Le schéma suivant décrit le principe de fonctionnement de la fonction AT, ainsi que son interaction avec les boucles PID :



Onglet AT de la fonction PID

L'écran suivant permet d'activer/désactiver la fonction AT et de paramétrer cette dernière.

Note : Il est accessible en mode local uniquement.



Description

⚠ AVERTISSEMENT

LA LIMITE DE LA VARIABLE DE RÉGULATION ET LES VALEURS DE CONSIGNE DE SORTIE DOIVENT ÊTRE DÉFINIES AVEC ATTENTION.

L'auto tuning de la fonction PID est un processus en boucle ouverte qui agit directement sur le processus de contrôle sans régulation ni autre limitation que celles définies par la limite de la mesure et la consigne de sortie. Vous devez donc sélectionner soigneusement les deux valeurs avec des valeurs comprises dans la plage autorisée, conformément au processus, et ce afin d'éviter toute éventuelle surcharge.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

⚠ AVERTISSEMENT

FONCTIONNEMENT INATTENDU DU MATERIEL

N'utilisez pas de sortie à relais avec un PID, car le nombre d'opérations de relais autorisé risquerait d'être dépassé et le relais d'être détruit. Selon le processus contrôlé, les conséquences peuvent être dangereuses.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
Autoriser	Cochez cette case si vous souhaitez activer le mode AT. Vous pouvez utiliser la case à cocher de deux façons, selon que vous définissez le mode de fonctionnement manuellement ou via une adresse mot dans l'onglet Général de la fonction PID : ● Si vous définissez le **Mode de fonctionnement** sur **AT+PID** ou sur **AT** dans l'onglet **Général** (reportez-vous à la section *p. 591*), l'option Autoriser est automatiquement cochée et indisponible (vous ne pouvez pas la désactiver). ● Si vous définissez le mode de fonctionnement via une adresse mot %MWx (%MWx = 2 : PID+AT ; %MWx = 3 : AT), vous devez cocher l'option Autoriser manuellement pour autoriser la configuration des paramètres d'auto tuning. **Résultat :** Dans ces deux cas, tous les champs de l'écran de configuration de l'onglet AT sont activés et vous devez saisir les valeurs adéquates dans les champs Consigne et Sortie.

Champ	Description
Limite de la variable de régulation	Spécifiez la limite de la variable de régulation au cours du processus d'auto tuning. Ce paramètre offre une sécurité au système de contrôle, l'auto tuning étant un processus en boucle ouverte. Cette valeur peut être un mot interne (%MW0 à un maximum de %MW2999, selon la quantité de mémoire système disponible), une constante interne (%KW0 à %KW255) ou une valeur directe. Cette valeur doit donc être comprise entre 0 et 10 000 lorsque la conversion est inhibée. Sinon, elle doit être comprise entre les valeurs minimum et maximum pour la conversion.
Consigne de sortie AT	Saisissez ici la valeur de la sortie d'auto tuning. Il s'agit de la valeur du changement d'étape appliquée au processus. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. La valeur doit donc être comprise entre 0 et 10 000. Vous devez sélectionner la valeur de consigne de sortie d'auto tuning de manière appropriée, en fonction de votre expérience pratique du processus régulé. En cas de doute ou pour déterminer la valeur la plus appropriée, sélectionnez le mode manuel et analysez la réponse du système suivant différents essais de consigne de sortie manuels. **Remarque :** La consigne de sortie d'auto tuning doit toujours être supérieure à la dernière sortie appliquée au processus.

Note : Lorsque la fonction AT est activée, les constantes (%KWx) ou les valeurs directes ne sont plus autorisées. Seuls les mots mémoire sont autorisés dans les champs PID suivants :

- Les paramètres **Kp**, **Ti** et **Td** doivent être définis en tant que **mots mémoire** (%MWx) dans l'onglet PID.
- Le champ **Action** est automatiquement défini sur **Bit adresse** dans l'onglet Sortie de la fonction PID.
- La case **Bit** doit être complétée par un **bit mémoire** (%Mx) approprié dans l'onglet Sortie.

Coefficients calculés Kp, Ti et Td

Lorsque le processus d'auto tuning est terminé, les coefficients PID calculés Kp, Ti et Td :

- sont stockés dans leurs mots mémoire (%MWx) respectifs ;
- apparaissent dans l'onglet **Animation**, en mode TwidoSuite connecté uniquement.

Onglet Sortie de la fonction PID

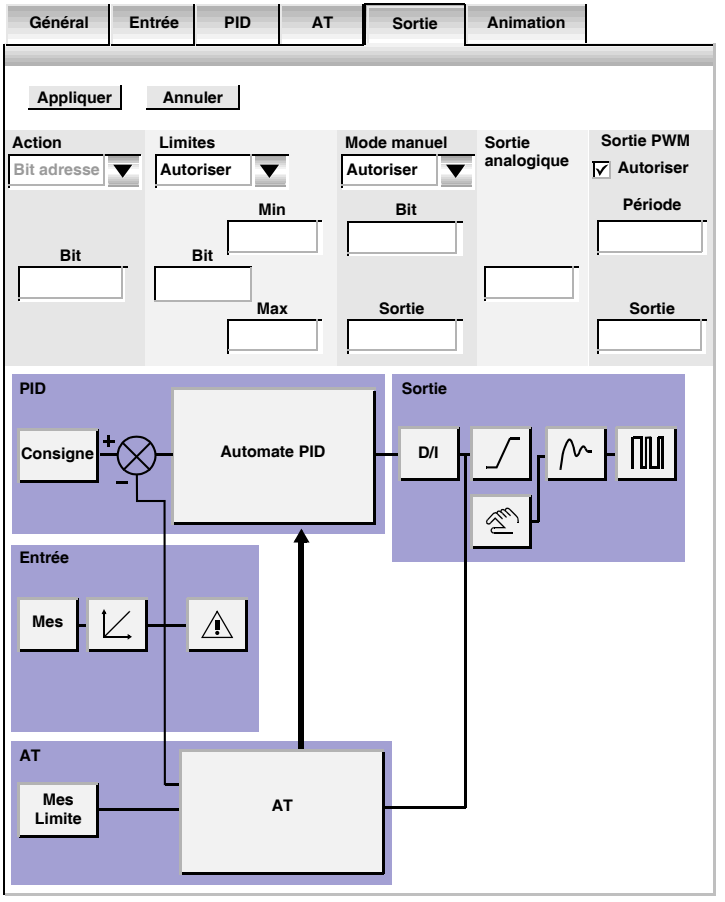
Présentation

Cet onglet permet de renseigner les paramètres de sortie de la fonction PID.

Note : Il est accessible en mode local.

Onglet Sortie de la fonction PID

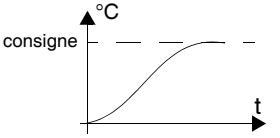
L'écran suivant permet de renseigner les paramètres internes du PID.



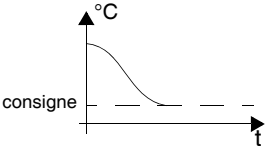
Description Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
PID numéro	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Action	Indiquez ici le type d'action de la fonction PID sur le processus. Trois options sont disponibles : **Inverse** , **Directe** ou **adresse bit**. Si vous avez sélectionné **adresse bit,** vous pouvez modifier ce type par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255), soit une entrée (%Ix.0 à %Ix.32). L'action est directe si le bit est à 1, et inverse dans le cas contraire. **Remarque :** Lorsque la fonction AT est activée, l'algorithme d'auto tuning détermine automatiquement le type d'action approprié (directe ou inverse) pour le processus de contrôle. Dans ce cas, une seule option est disponible dans la liste déroulante Action : **Bit adresse.** Vous devez ensuite saisir dans la zone de texte du **Bit** associé, un **mot interne** (%MW0 à %MW2999). Ne tentez pas de saisir une constante interne ou une valeur directe dans la zone de texte **Bit**, car cela déclenche une erreur d'exécution.
Limites Bit	Indiquez ici si vous désirez limiter la sortie de la fonction PID. Trois options sont disponibles : **Autoriser**, **Inhiber** ou **adresse bit**. Si vous avez sélectionné **adresse bit,** vous pouvez autoriser (bit à 1) ou inhiber (bit à 0) la gestion des limites par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255), soit une entrée (%Ix.0 à %Ix.32).
Min. **Max.**	Définissez ici les limites haute et basse pour la sortie du PID. **Remarque** : La valeur **Mini** doit obligatoirement être inférieure à la valeur **Maxi**. **Min.** ou **Max.** peuvent être soit des mots internes (%MW0 à %MW2999), soit des constantes internes (%KW0 à %KW255), soit une valeur comprise entre 1 et 10 000.
Mode manuel Bit **Sortie**	Indiquez ici si vous désirez activer le mode manuel pour la fonction PID. Trois options sont disponibles : **Autoriser**, **Inhiber** ou **adresse bit**. Si vous avez sélectionné **adresse bit,** vous pouvez passer en mode manuel (bit à 1) ou en mode automatique (bit à 0) par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255), soit une entrée (%Ix.0 à %Ix.32). La **Sortie** du mode manuel doit contenir la valeur que vous désirez affecter à la sortie analogique lorsque le PID est en mode manuel. Cette **Sortie** peut être soit un mot (%MW0 à %MW2999), soit une valeur directe au format [0-10 000].
Sortie analogique	Indiquez ici la sortie de la fonction PID en mode automatique. Cette **Sortie analogique** peut être de type %MW (%MW0 à %MW2999) ou %QW (%QWx.0).
Sortie PWM activée **Période (0,1 s)** **Sortie**	Cochez la case si vous souhaitez utiliser la fonction PWM de PID. Spécifiez la période de modulation dans **Période (0,1 s)**. Cette période doit être comprise entre 1 et 500. Elle peut être un mot interne (%MW0 à %MW2999) ou une constante interne (%KW0 à %KW255). Indiquez dans la valeur **Sortie**le bit de sortie PWM. Il peut s'agir d'un bit interne (%M0 à %M255) ou d'une sortie (%Qx.0 à %Qx.32).
Synoptique	Le synoptique vous permet de visualiser les différentes configurations possibles du PID.

Note : Le terme Inverse dans le champ Action est utilisé pour atteindre une consigne haute (ex. : pour chauffer)



Le terme Directe dans le champ Action est utilisé pour atteindre une consigne basse (ex. : pour refroidir)



Comment accéder à la mise au point du PID

Présentation L'accès aux écrans de mise au point d'un PID sur automates TWIDO est décrit dans les paragraphes qui suivent.

Marche à suivre Le tableau suivant présente la marche à suivre pour accéder aux écrans de mise au point d'un PID :

Etape	Action
1	Vérifiez que vous êtes en mode **connecté**.
2	Dans l'écran de configuration logicielle du moniteur, sélectionnez **Objets avancés** dans la fenêtre Catégories d'objets (Fenêtre Catégories d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne) et **PID** dans la fenêtre Type d'objets (Fenêtre des types d'objets, Twido, Logiciel de programmation TwidoSuite, Aide en ligne).
3	Sélectionnez le PID souhaité (champ **PID numéro**) dans le tableau des PID. **Remarque** : Vous pouvez également double-cliquer sur l'élément graphique PID dans le réseau langage schéma à contacts pour accéder à la fenêtre de configuration de l'automate PID.
4	Cliquez sur l'onglet **Animation**. **Résultat :** la fenêtre de paramétrage des PID s'ouvre, elle est par défaut positionnée sur l'onglet **Animation** (voir *p. 607*).

Onglet Animation de la fonction PID

Présentation

Cet onglet permet d'effectuer la mise au point du PID.

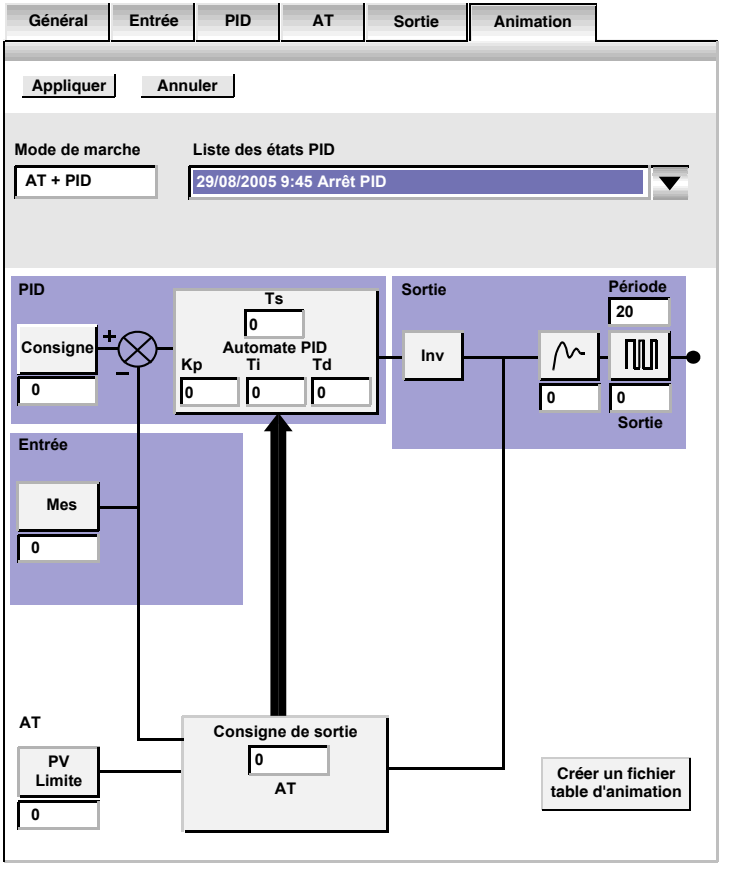
Le synoptique affiché dépend du type de PID que vous avez créé, seuls les éléments configurés apparaissent.

La visualisation est dynamique, les liaisons actives apparaissent en rouge, les liaisons inactives apparaissent en noir.

Note : Il est accessible en mode connecté.

Onglet Animation du PID

L'écran suivant permet de visualiser et de mettre au point le PID.



Description Le tableau suivant décrit les différentes zones de la fenêtre.

Champ	Description
Numéro PID	Indiquez ici le numéro du PID que vous désirez mettre au point. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Mode de marche	Ce champ affiche le mode de marche courant du PID.
Liste des états PID	Cette liste déroulante permet de visualiser en temps réel les 15 derniers états du PID. Chaque changement d'état met à jour cette liste en indiquant la date et l'heure ainsi que l'état courant.
Créer un fichier table d'animation	Cliquez sur le bouton **Créer un fichier table d'animation**, pour créer un fichier contenant toutes les variables visualisées sur le synoptique afin de vous permettre de les modifier en mode connecté et d'effectuer la mise au point de votre PID.

Ecran Trace de la fonction PID

Présentation

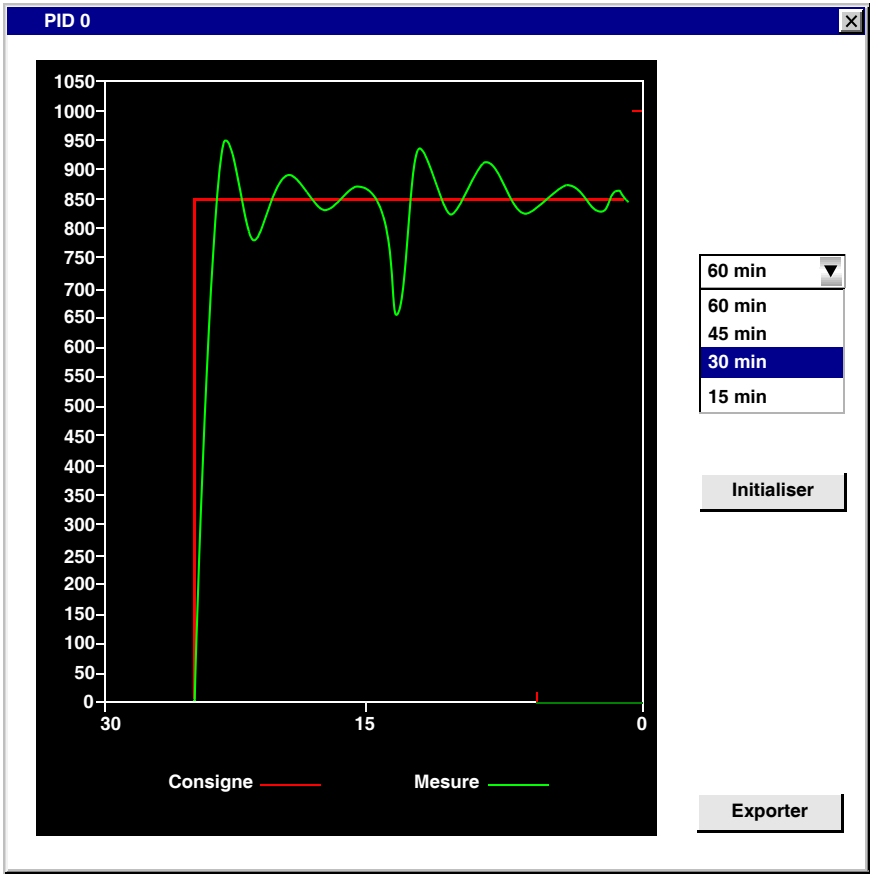
Cet écran permet de visualiser le fonctionnement du PID et d'effectuer ses réglages.

Le tracé des courbes débute dès l'affichage de la fenêtre de mise au point.

Note : Il est accessible en mode connecté.

Onglet Animation du PID

L'écran suivant permet de visualiser la régulation du PID.



Description Le tableau suivant décrit les différentes zones de la fenêtre.

Champ	Description
PID numéro	Indiquez ici le numéro du PID que vous désirez visualiser. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
Graphe	Cette zone affiche les courbes de la **consigne** et de la **mesure**. L'échelle selon l'axe horizontal (X) est déterminée par le menu situé en haut à droite. L'échelle selon l'axe vertical est déterminé par les valeurs de configuration de l'entrée du PID (avec ou sans conversion). Elle est automatiquement optimisée afin de visualiser au mieux les courbes.
Menu d'échelle axe horizontal	Ce menu permet de modifier l'échelle horizontale. Vous pouvez choisir entre 4 valeurs : 15, 30, 45 ou 60 minutes.
Initialiser	Ce bouton efface la courbe et relance la visualisation des tracés.
Exporter	Ce bouton vous permet d'exporter les données d'affichage au format Excel. Cliquez sur Exporter pour ouvrir une boîte de dialogue dans laquelle vous pouvez spécifier le nom et l'emplacement d'un fichier *.cvs*. Dans cette boîte de dialogue, cliquez sur Enregistrer pour exporter les données ou sur Annuler pour annuler l'export.

Etats du PID et codes d'erreurs

Présentation

Outre la **liste des états du PID** disponible dans la boîte de dialogue **Animation** (voir *p. 607*) qui permet de visualiser et commuter vers l'un des 15 derniers états PID, l'automate PID Twido a également la capacité d'enregistrer l'état courant de l'automate PID et du processus AT dans un mot mémoire défini par l'utilisateur.

Pour savoir comment activer et configurer le **mot mémoire de l'état du PID** (%MWi), reportez-vous *p. 591*.

Mot mémoire de l'état du PID

Le mot mémoire de l'état du PID peut enregistrer l'un des trois types d'informations du PID, comme suit :

- Etat courant de l'automate PID (état du PID)
- Etat courant du processus d'auto tuning (état AT)
- PID et codes d'erreurs AT

Note : Le mot mémoire de l'état PID est en lecture seule.

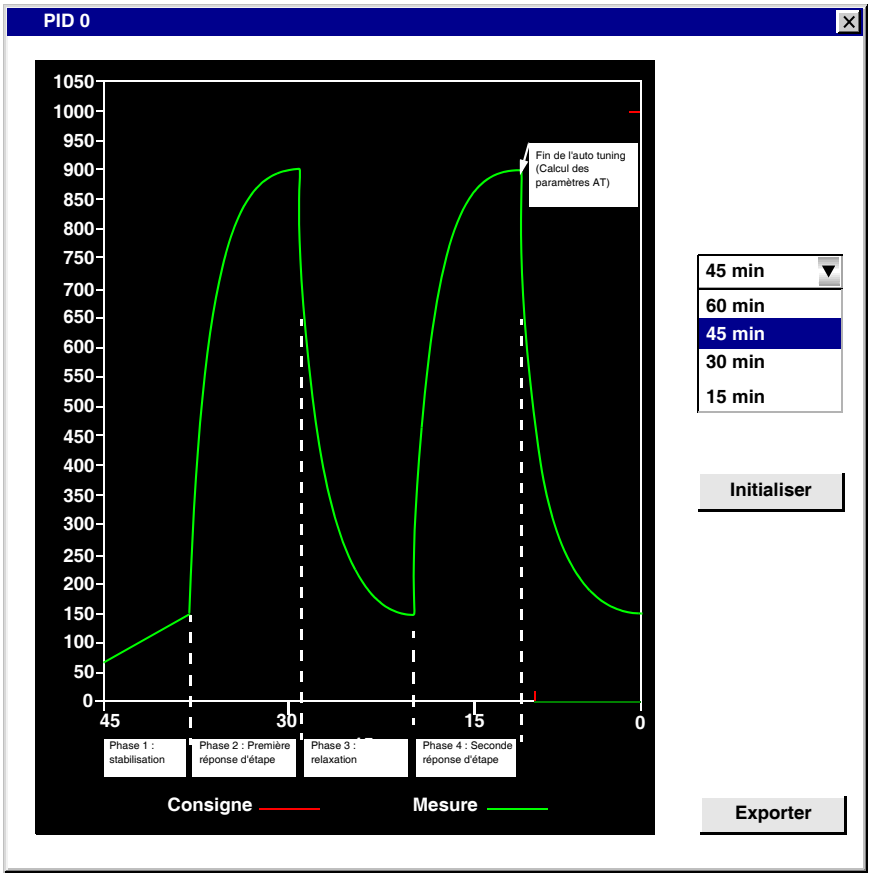
Mot mémoire de l'état du PID

Le tableau de concordance de codage hexadécimal suivant indique l'état de l'automate PID par rapport au mot mémoire :

Notation hexadécimale de l'état PID	Description
0000h	Le contrôle PID est inactif
2000h	Le contrôle PID est en cours
4000h	La consigne PID est atteinte

Description de l'état AT

Le processus d'auto tuning se divise en 4 phases consécutives. Chaque phase du processus doit être réalisée pour mener à bien l'auto tuning. La courbe de réponse du processus suivante et le tableau décrivent les 4 phases de l'auto tuning PID Twido :



Les phases d'auto tuning sont décrites dans le tableau suivant :

Phase AT	Description
1	La **Phase 1** est la phase de stabilisation. Elle commence lorsque l'utilisateur lance le processus AT. Au cours de cette phase, l'auto tuning de Twido effectue des vérifications pour garantir que l'état de la variable du processus est stable. **Remarque :** La dernière sortie appliquée au processus avant le début de l'auto tuning est utilisée, tout comme le point de démarrage et le point de relaxation pour le processus d'auto tuning.
2	La **Phase 2** applique le premier changement d'étape au processus. Elle génère une réponse d'étape au processus similaire à celle illustrée dans le schéma ci-dessus.
3	La **Phase 3** est la phase de relaxation qui débute lorsque la première réponse d'étape est stabilisée. **Remarque :** La relaxation se produit vers l'équilibre qui est déterminé comme la dernière sortie appliquée au processus avant le démarrage de l'auto tuning.
4	La **Phase 4** applique le deuxième changement d'étape au processus avec le même montant et de la même manière que dans la phase 2 décrite ci-dessus. Le processus d'auto tuning se termine et les paramètres AT sont calculés, puis enregistrés dans leurs mots mémoire respectifs une fois la phase 4 réalisée. **Remarque :** Une fois cette phase exécutée, la variable du processus est restaurée dans le dernier niveau de sortie appliqué au processus avant le démarrage de l'auto tuning.

Mot mémoire de l'état AT

Le tableau de concordance de codage hexadécimal suivant indique l'état de l'automate PID par rapport au mot mémoire :

Notation hexadécimale de l'état AT	Description
0100h	Phase 1 d'auto tuning en cours
0200h	Phase 2 d'auto tuning en cours
0400h	Phase 3 d'auto tuning en cours
0800h	Phase 4 d'auto tuning en cours
1000h	Processus d'auto tuning terminé

PID et codes d'erreurs AT

Le tableau suivant décrit les erreurs d'exécution potentielles pouvant survenir au cours des processus de contrôle PID et d'auto tuning :

Processus PID/AT	Code d'erreur (hexadécimal)	Description
Erreur PID	8001h	Valeur du mode de fonctionnement hors plage
	8002h	Les valeurs minimum et maximum de la conversion linéaire sont égales
	8003h	La limite supérieure pour la sortie TOR est inférieure à la limite inférieure
	8004h	La limite de la variable du processus se trouve en dehors de la plage de conversion linéaire
	8005h	La limite de la variable du processus est inférieure à 0 ou supérieure à 10000
	8006h	La consigne se trouve en dehors de la plage de conversion linéaire
	8007h	La consigne est inférieure à 0 ou supérieure à 10000
	8008h	L'action du contrôle est différente de l'action déterminée au démarrage AT
Erreur d'auto tuning	8009h	Erreur d'auto tuning : la limite de la variable du processus est atteinte
	800Ah	Erreur d'auto tuning : due à un sur-échantillonnage ou à une consigne de sortie trop faible.
	800Bh	Erreur d'auto tuning : Kp est égal à zéro
	800Ch	Erreur d'auto tuning : la constante de temps est négative
	800Dh	Erreur d'auto tuning : le retard est négatif
	800Eh	Erreur d'auto tuning : erreur de calcul du paramètre Kp
	800Fh	Erreur d'auto tuning : constante de temps supérieure au délai > 20.
	8010h	Erreur d'auto tuning : constante de temps supérieure au délai < 2.
	8011h	Erreur d'auto tuning : la limite du paramètre Kp est dépassée
	8012h	Erreur d'auto tuning : la limite du paramètre Ti est dépassée
	8013h	Erreur d'auto tuning : la limite du paramètre Td est dépassée

Réglage PID avec la fonction d'auto tuning (AT)

Vue d'ensemble du réglage PID

La fonction du contrôle PID repose sur les trois paramètres suivants définis par l'utilisateur : Kp, Ti et Td. Le réglage PID vise à déterminer de manière précise ces paramètres de processus pour offrir un contrôle optimal du processus.

Objectif de l'auto tuning

La fonction AT de l'automate Twido est spécifiquement adaptée au réglage automatique des processus thermiques. Etant donné que les valeurs des paramètres PID peuvent varier considérablement d'une régulation à une autre, la fonction d'auto tuning fournie par l'automate Twido peut vous aider à déterminer des valeurs plus précises que celles basées sur vos hypothèses et ce, avec moins d'effort.

Conditions de l'auto tuning

Lors de l'utilisation de la fonction d'auto tuning, assurez-vous que la régulation et l'automate Twido respectent les quatre exigences suivantes :

- La régulation doit être un système stable à boucle ouverte.
- Au début de l'exécution d'auto tuning, la régulation doit être dans un état stable avec une entrée de processus nulle (par exemple, un four ou un fourneau doit être à température ambiante).
- Lors du fonctionnement de l'auto tuning, veillez à ce qu'aucune perturbation ne s'introduise dans le processus. Sinon, les paramètres calculés seront erronés ou le processus d'auto tuning échouera (par exemple, la porte du four ne doit pas être ouverte, même momentanément).
- Configurez l'automate Twido afin qu'il scrute en **mode périodique.** Une fois que vous avez déterminé la période d'échantillonnage correcte (Ts) pour l'auto tuning, la période de scrutation doit être configurée pour que la période d'échantillonnage (Ts) soit un multiple exact de la période de scrutation de l'automate Twido.

Note : Pour garantir une bonne exécution du contrôle PID et du processus d'auto tuning, il est essentiel de configurer l'automate Twido pour exécuter les scrutations en mode périodique (et non cyclique). En mode périodique, chaque scrutation de l'automate débute à des intervalles réguliers. Ainsi, le taux d'échantillonnage est constant tout au long de la mesure (contrairement au mode cyclique où une scrutation commence dès que la précédente est terminée, ce qui crée un déséquilibre au niveau de la période d'échantillonnage d'une scrutation à l'autre).

Modes de marche AT

L'auto tuning peut être utilisé indépendamment (mode AT) ou conjointement avec le contrôle PID (AT + PID) :

- **Mode AT :** Après la convergence du processus AT et une détermination réussie des paramètres Kp, Ti et Td du contrôle PID (ou après la détection d'une erreur dans l'algorithme AT), la sortie numérique AT est réglée sur 0 et le message suivant apparaît dans la liste déroulante **Liste des états du PID** : "Auto tuning terminé".
- **Mode AT + PID :** L'auto tuning est lancé en premier. Après l'exécution réussie de l'auto tuning, la boucle du contrôle PID démarre (en fonction des paramètres Kp, Ti et Td calculés par l'auto tuning).
  Remarque sur AT+PID : Si une erreur survient dans l'algorithme AT :
  - aucun paramètre PID n'est calculé ;
  - la sortie numérique AT est réglée sur la dernière sortie appliquée au processus avant le démarrage de l'auto tuning ;
  - un message d'erreur apparaît dans la liste déroulante Liste des états du PID ;
  - le contrôle PID est annulé.

Note : Transition sans à-coups
En **mode AT+PID**, la transition de AT à PID est sans à-coups.

Méthodes pour déterminer la période d'échantillonnage (Ts)

Comme il sera expliqué dans les deux sections suivantes (voir *p. 630* et *p. 632*), la **période d'échantillonnage (Ts)** est un paramètre clé du contrôle PID. La période d'échantillonnage peut être déduite de la **constante temps AT (τ).**

Il existe deux méthodes pour évaluer la période d'échantillonnage correcte (Ts) à l'aide de la fonction d'auto tuning. Ces méthodes sont décrites dans les sections suivantes.

- La méthode de la courbe de réponse du processus
- La méthode des essais et erreurs

Ces deux méthodes sont décrites dans les deux sous-sections suivantes.

Présentation de la méthode de la courbe de réponse du processus

Cette méthode consiste à configurer un changement d'étape à l'entrée de régulation et à enregistrer la courbe de sortie du processus par rapport au temps.

La méthode de la courbe de réponse du processus suppose que :

- La régulation peut être décrite de manière adéquate en tant que condition de premier ordre avec modèle de temporisation par la fonction de transfert suivante :

$$\frac{S}{U} = \frac{k}{1+\tau p} \cdot e^{-\theta p}$$

(Reportez-vous à l'annexe 2 pour obtenir davantage d'informations : Premier ordre avec modèle de temporisation)

Utilisation de la méthode de la courbe de réponse du processus

Pour déterminer la période d'échantillonnage (Ts) à l'aide de la méthode de la courbe de réponse du processus, procédez comme suit :

Etape	Action
1	Divers réglages doivent déjà être effectués dans les onglets Général, Entrée, PID, AT et Sortie du PID.
2	Sélectionnez l'onglet **PID > Sortie**.
3	Sélectionnez **Autoriser** ou **Bit adresse** dans la liste déroulante **Mode manuel** pour autoriser la sortie manuelle et définir le champ **Sortie** sur un niveau élevé (dans la plage [5 000 - 10 000]).
4	Sélectionnez **Automate > Transfert PC => Automate...** pour télécharger le programme d'application de l'automate Twido.
5	Dans la fenêtre de configuration PID, passez en mode **Trace**.
6	Exécutez le PID et vérifiez l'augmentation de la courbe de réponse.
7	Lorsque la courbe de réponse a atteint un état stable, arrêtez la mesure PID. **Remarque :** Gardez la fenêtre PID Trace active.
8	Utilisez la méthode graphique suivante pour déterminer la constante de temps (τ) de la régulation : **1.** Calculez la sortie de la variable du processus pour une augmentation de 63 % ($S_{[63\ \%]}$) en utilisant la formule suivante : $S_{[63\ \%]} = S_{[initial]} + (S_{[final]} - S_{[initial]}) \times 63\ \%$ **2.** Repérez sur le graphique l'abscisse du temps ($t_{[63\ \%]}$) qui correspond à S (63 %). **3.** Repérez sur le graphique le temps initial ($t_{[initial]}$) qui correspond au début de l'augmentation de la réponse du processus. **4.** Calculez la constante de temps (τ) de la régulation en utilisant la relation suivante : $\tau = t_{[63\ \%]} - t_{[initial]}$
9	Calculez la période d'échantillonnage (Ts) en fonction de la valeur de (τ) que vous avez déterminée à l'étape précédente, en utilisant la règle suivante : $Ts = \tau/75$ **Remarque :** L'unité de base de la période d'échantillonnage est de 10 ms. Par conséquent, vous devez arrondir la valeur Ts aux 10 ms supérieures ou inférieures.
10	Sélectionnez la tâche **Programme > Configurer le comportement** pour régler les paramètres du **Mode de scrutation** et procédez comme suit : **1.** Définissez le **mode de scrutation** de l'automate Twido sur **Périodique**. **2.** Définissez la **période de scrutation** de façon à ce que la période d'échantillonnage (Ts) soit un **multiple exact** de la période de scrutation, en utilisant la règle suivante : Période de scrutation = Ts / n, où "n" est un entier positif. **Remarque :** Vous devez choisir n pour que la période de scrutation résultante soit un entier positif dans la plage [2-150ms].

Exemple de courbe de réponse du processus

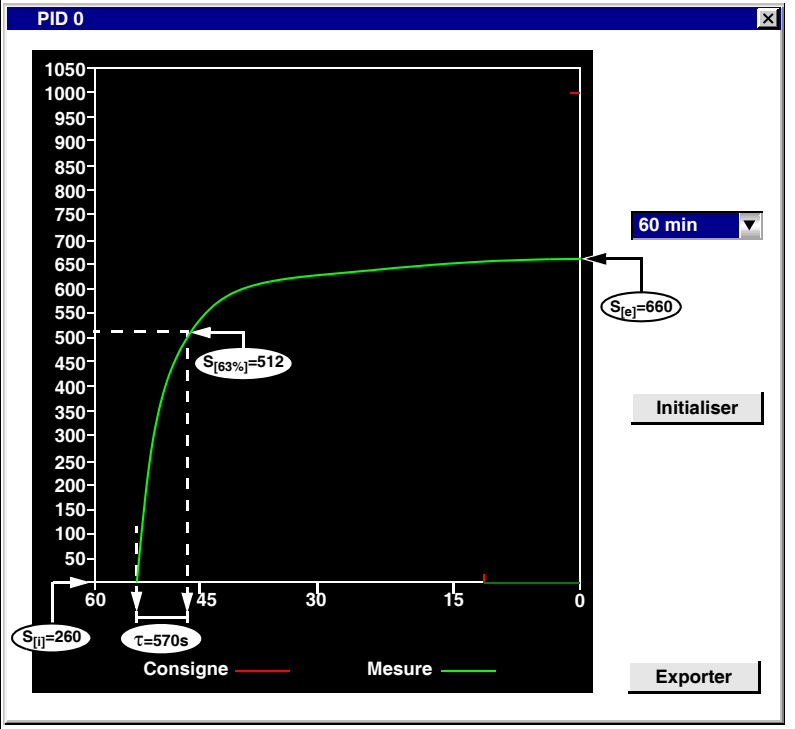
Cet exemple vous montre comment mesurer la constante de temps (τ) d'un processus thermique simple à l'aide de la méthode de la courbe du processus décrite dans la sous-section précédente.

Le paramétrage expérimental de la mesure de la constante du temps est le suivant :

- La régulation consiste en un four à air forcé équipé d'un témoin de marche.
- Les mesures de température sont regroupées par l'automate Twido via une sonde Pt100 et la température est enregistrée en °C.
- L'automate Twido contrôle un témoin de marche via la sortie TOR PWM du PID.

L'expérience est réalisée comme suit :

Etape	Action
1	L'onglet **Sortie** du PID est sélectionné dans la fenêtre de configuration PID.
2	**Le mode manuel** est sélectionné dans l'onglet Sortie.
3	Le mode manuel **Sortie** est réglé sur 10 000.
4	Le PID est exécuté depuis l'onglet **Trace PID.**
5	L'exécution du PID est arrêtée lorsque la température du four est stationnaire.

Etape	Action
6	Les informations suivantes proviennent directement de l'analyse graphique de la courbe de réponse, comme illustré dans le schéma ci-dessous :  Où ● $S_{[i]}$ = valeur initiale de la variable du processus = 260 ● $S_{[e]}$ = valeur finale de la variable du processus = 660 ● $S_{[63\ \%]}$ = variable du processus à une augmentation de 63 % = $S_{[i]} + (S_{[i]} - S_{[e]}) \times 63\ \%$ = 260 + (660 - 260) x 63 % = 512 ● τ = constante de temps = temps écoulé depuis le début de l'augmentation jusqu'à ce $S_{[63\ \%]}$ soit atteint. = 9 min 30 s = 570 s
7	La période d'échantillonnage (Ts) est déterminée à l'aide de la relation suivante : Ts = τ/75 = 570/75 = 7,6 s (7 600 ms)
8	Dans la boîte de dialogue **Programme > Editer le mode de scrutation**, la **période de scrutation** doit être définie de sorte que la période d'échantillonnage (Ts) est le multiple exact de la période de scrutation, comme dans l'exemple suivant : Période de scrutation = Ts/76 = 7 600/76 = 100 ms (qui satisfait la condition : 2 ms ≤Période de scrutation ≤ 150 ms.)

Méthode des essais et erreurs

La méthode des essais et erreurs consiste à fournir des hypothèses successives de la période d'échantillonnage à la fonction d'auto tuning jusqu'à ce que l'algorithme réussisse à converger vers les paramètres Kp, Ti et Td qui sont jugés satisfaisants par l'utilisateur.

Note : Contrairement à la méthode de courbe de réponse au processus, la méthode essai et erreur n'est basée sur aucune loi d'approximation de la réponse au processus. Cependant, elle a l'avantage de pouvoir converger vers une valeur de la période d'échantillonnage se trouvant dans le même ordre de grandeur que la valeur actuelle.

Pour effectuer une estimation des essais et erreurs des paramètres d'auto tuning, procédez comme suit :

Etape	Action
1	Sélectionnez l'onglet **AT** dans la fenêtre de configuration PID.
2	Paramétrez la **limite de sortie** d'auto tuning sur **10 000**.
3	Sélectionnez l'onglet **PID** dans la fenêtre de configuration PID.
4	Saisissez la première ou la n^{ième} hypothèse dans le champ **Période d'échantillonnage**. **Remarque :** Si vous ne possédez pas de première indication sur la plage possible de la période d'échantillonnage, définissez cette valeur sur la plus petite possible: 1 (1 unité de 10 ms).
5	Sélectionnez **Automate > Transfert PC => Automate...** dans la barre de menus pour télécharger le programme d'application de l'automate Twido.
6	Lancez l'**auto tuning.**
7	Sélectionnez l'onglet **Animation** dans l'écran de configuration PID.
8	Patientez jusqu'à la fin du processus d'auto tuning.
9	Deux cas peuvent se produire : ● **L'auto tuning se termine avec succès :** Vous pouvez continuer jusqu'à l'étape 9. ● **L'auto tuning échoue :** Ceci signifie que la supposition courante pour la période d'échantillonnage (Ts) n'est pas correcte. Essayez une nouvelle hypothèse Ts et répétez les étapes 3 à 8, autant de fois que nécessaire jusqu'à ce que le processus d'auto tuning converge. Suivez ces instructions pour fournir une nouvelle supposition Ts : ● L'auto tuning se termine en indiquant le message d'erreur **"La constante de temps calculée est négative." :** ceci signifie que la période d'échantillonnage **Ts est trop importante.** Diminuez la valeur Ts pour fournir une nouvelle supposition. ● L'auto tuning se termine en indiquant le message d'erreur **"Erreur d'échantillonnage." :** ceci signifie que la période d'échantillonnage **Ts est trop petite.** Augmentez la valeur Ts pour fournir une nouvelle supposition.
10	Il se peut que vous ne puissiez pas visualiser les paramètres du contrôle PID (Kp, Ti et Td) dans l'onglet Animation. Ajustez-les dans l'onglet PID de l'écran de configuration comme il convient. **Remarque :** Si la régulation du PID fournie par cet ensemble de paramètres de contrôle n'indique pas des résultats totalement satisfaisants, vous pouvez affiner l'évaluation des essais et erreurs de la période d'échantillonnage jusqu'à l'obtention d'un ensemble approprié des paramètres de contrôle Kp, Ti et Td.

Ajustement des paramètres PID

Pour affiner la régulation du processus fournie par les paramètres PID (Kp, Ti, Td) provenant de l'auto tuning, vous pouvez également ajuster manuellement la valeur des paramètres, directement à partir de l'onglet PID de l'écran de configuration PID ou via les mots mémoire correspondants (%MW).

Limites à l'utilisation de l'auto tuning et du contrôle PID

L'**auto tuning** convient particulièrement aux processus dont la constante de temps (τ) et le temps de retard (θ) respectent l'exigence suivante : 10 s < (τ + θ) < 2 700 s (ex. : 45 min)

Note : L'auto tuning ne fonctionne pas dans les cas suivants : (τ/θ) < 2 ou (τ/θ) > 20.

Le **contrôle PID** convient particulièrement à la régulation des processus qui satisfont à la condition suivante : 2 < (τ/θ) < 20, où (τ) est la constante de temps du processus et (θ) le temps de retard.

Note : Selon le rapport (τ/θ) :
- (τ/θ) < 2 : la régulation PID a atteint ses limites. Des techniques de régulation plus avancées sont requises dans ce cas.
- (τ/θ) > 20 : dans ce cas, un simple automate allumé/éteint (ou à deux étapes) peut être utilisé à la place de l'automate PID.

Résolution des erreurs de la fonction d'auto tuning

Le tableau suivant enregistre les messages d'erreur d'auto tuning et décrit les causes possibles, ainsi que les actions à prendre en matière de résolution :

Message d'erreur	Cause possible	Explication / Solution possible
Erreur d'auto tuning : la limite de la variable du processus est atteinte.	La variable du processus a atteint la valeur maximale autorisée.	Est utile pour la sécurité du système. L'auto tuning étant un processus à boucle ouverte, la limite de la variable du processus fonctionne comme une limite supérieure.
Erreur d'auto tuning : due à un sur-échantillonnage ou à une consigne de sortie trop faible.	Deux causes possibles : • la période d'échantillonnage est trop petite ; • le réglage de la sortie d'auto tuning est trop faible.	Augmentez la période d'échantillonnage ou la valeur de consigne de sortie d'auto tuning.
Erreur d'auto tuning : la constante de temps est négative.	La période d'échantillonnage est peut-être trop importante.	Pour plus d'informations, consultez la section *p. 615*.

Message d'erreur	Cause possible	Explication / Solution possible
Erreur d'auto tuning : erreur de calcul du paramètre Kp.	L'algorithme AT a échoué (pas de convergence). • Des perturbations pendant l'auto tuning ont causé une distorsion lors du calcul du gain statique. • La réponse transitoire de la variable du processus est insuffisante pour que l'auto tuning puisse calculer le gain statique. • Une combinaison des causes ci-dessous peut influer sur le processus.	Vérifiez les paramètres PID et AT et effectuez des ajustements pour améliorer la convergence. Vérifiez également qu'aucune perturbation n'affecte la variable du processus. Essayez de modifier : • la consigne de sortie ; • la période d'échantillonnage. Assurez-vous qu'il n'y a aucune perturbation lorsque l'auto tuning est en cours.
Erreur auto tuning : constante de temps supérieure au délai > 20.	$\tau/\theta > 20$	La régulation PID n'est plus garantie. Pour plus d'informations, consultez la section *p. 615*.
Erreur d'auto tuning : constante de temps supérieure au délai < 2	$\tau/\theta < 2$	La régulation PID n'est plus garantie. Pour plus d'informations, consultez la section *p. 615*.
Erreur d'auto tuning : la limite du paramètre Kp est dépassée.	La valeur calculée du gain statique (Kp) est supérieure à 10 000.	La sensibilité de la mesure de certaines variables d'application est peut-être trop faible. La plage de mesure de l'application doit être réévaluée dans l'intervalle [0 - 10 000].
Erreur d'auto tuning : la limite du paramètre Ti est dépassée.	La valeur calculée de la constante de temps intégral (Ti) est supérieure à 20 000.	La limite de calcul est atteinte.
Erreur d'auto tuning : la limite du paramètre Td est dépassée.	La valeur calculée de la constante de temps dérivative (Td) est supérieure à 10 000.	La limite de calcul est atteinte.

Méthode de réglage du paramètre PID

Présentation

De nombreuses méthodes permettent de régler les paramètres PID. Nous vous recommandons les méthodes Ziegler et Nichols qui présentent deux variantes :

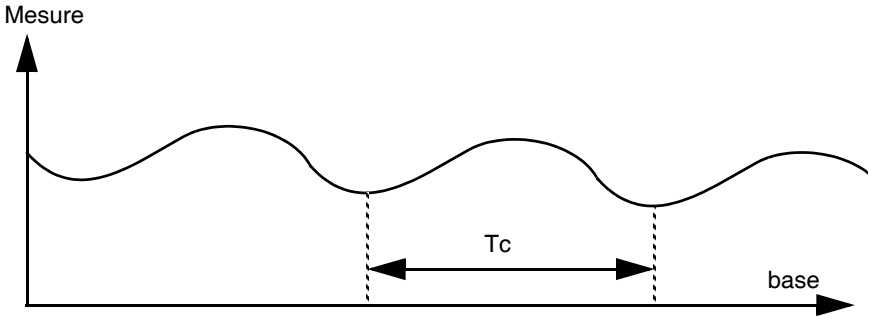
- réglage de boucle fermée ;
- réglage de boucle ouverte.

Avant d'appliquer l'une de ces méthodes, vous devez définir la direction de l'action PID :

- si une augmentation de la sortie OUT entraîne une augmentation de la mesure PV, inversez le PID (KP > 0) ;
- d'autre part, si ceci entraîne une réduction de la mesure PV, faites en sorte que le PID soit direct (KP < 0).

Réglage de boucle fermée

Ce principe consiste à utiliser une commande proportionnelle (Ti = 0, Td = 0) pour démarrer le processus, en augmentant la production jusqu'à ce que l'oscillation recommence après application d'un niveau à la consigne du correcteur PID. Il suffit d'augmenter le niveau de production critique (Kpc) qui a entraîné l'oscillation non amortie, et d'augmenter la période d'oscillation (Tc) pour réduire les valeurs, permettant ainsi une régulation optimale du régulateur.



Selon le type de régulateur (PID ou PI), le réglage des coefficients est effectué avec les valeurs suivantes :

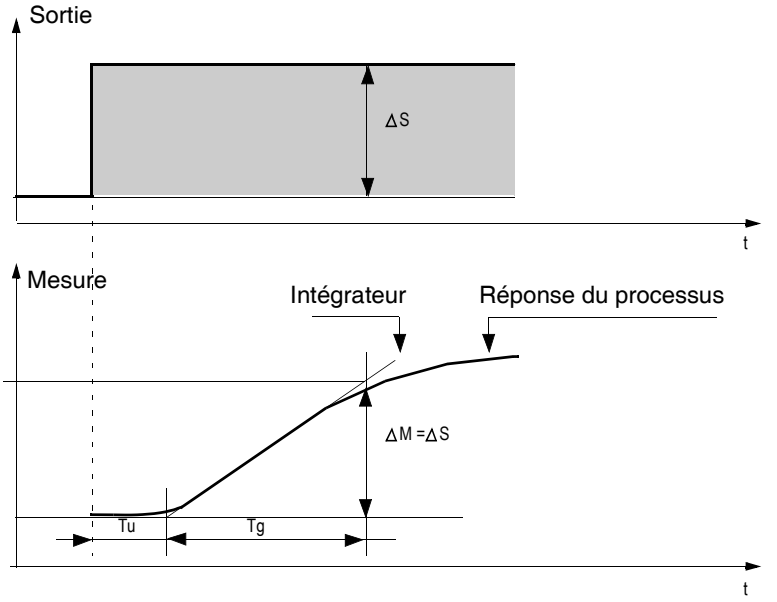
-	Kp	Ti	Td
PID	Kpc/1,7	Tc/2	Tc/8
PI	Kpc/2,22	0,83 x Tc	-

où Kp = production proportionnelle, Ti = temps d'intégration et TD = temps de diversion.

Note : Cette méthode de réglage fournit une commande particulièrement dynamique qui peut s'exprimer par des dépassements non souhaités lors du changement d'impulsions de consigne. Dans ce cas, baissez la valeur de production jusqu'à obtenir le comportement requis.

Réglage de boucle ouverte

Lorsque le régulateur est en mode manuel, vous appliquez un niveau à la sortie et vous lancez la procédure de réponse comme pour un intégrateur avec un temps de retard pur.



Le point d'intersection sur le côté droit, représentant l'intégrateur avec les axes de temps, détermine le temps Tu. Le temps Tg est ensuite défini comme le temps nécessaire pour que la variable contrôlée (mesure) ait la même taille de variation (% de l'échelle) que la sortie du régulateur.

Selon le type de régulateur (PID ou PI), le réglage des coefficients est effectué avec les valeurs suivantes :

-	Kp	Ti	Td
PID	-1,2 Tg/Tu	2 x Tu	0,5 x Tu
PI	-0,9 Tg/Tu	3,3 x Tu	-

où Kp = production proportionnelle, Ti = temps d'intégration et TD = temps de diversion.

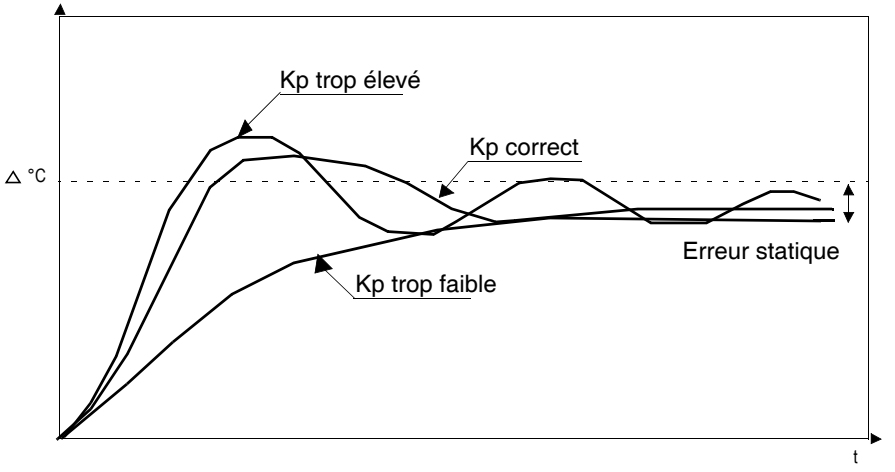
Note : Attention aux unités. Si le réglage est effectué dans l'automate PL7, multipliez la valeur obtenue pour KP par 100.

Cette méthode de réglage fournit également une commande particulièrement dynamique qui peut s'exprimer par des dépassements non souhaités lors du changement d'impulsions de consigne. Dans ce cas, baissez la valeur de production jusqu'à obtenir le comportement requis. Cette méthode est intéressante, car elle ne requiert aucune hypothèse sur la nature et l'ordre de la procédure. Vous pouvez l'appliquer aussi bien aux procédures stables qu'aux procédures d'intégration réelles. Elle est particulièrement intéressante dans le cas de procédures lentes (industrie du verre,…), car l'utilisateur a uniquement besoin du début de la réponse pour régler les coefficients Kp, Ti et Td.

Rôle et influence des paramètres PID

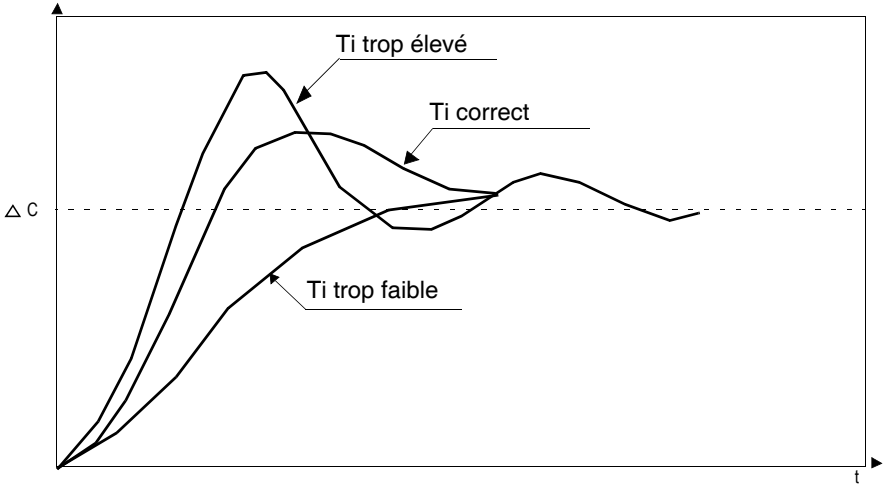
Influence de l'action proportionnelle

L'action proportionnelle sert à influencer la vitesse de réponse du processus. Plus le gain est élevé, plus la réponse est rapide et plus l'erreur statique est faible (en proportion directe), mais aussi plus la stabilité se détériore. Il convient de trouver un compromis approprié entre la vitesse et la stabilité. L'influence de l'action intégrale sur la réponse du processus à une division d'échelle est la suivante :



Influence de l'action intégrale

L'action intégrale sert à annuler l'erreur statique (écart entre la mesure et la consigne). Plus le niveau d'action intégrale est élevé (Ti faible), plus la réponse est rapide et plus la stabilité se détériore. Il convient également de trouver un compromis approprié entre la vitesse et la stabilité. L'influence de l'action intégrale sur la réponse du processus à une division d'échelle est la suivante:

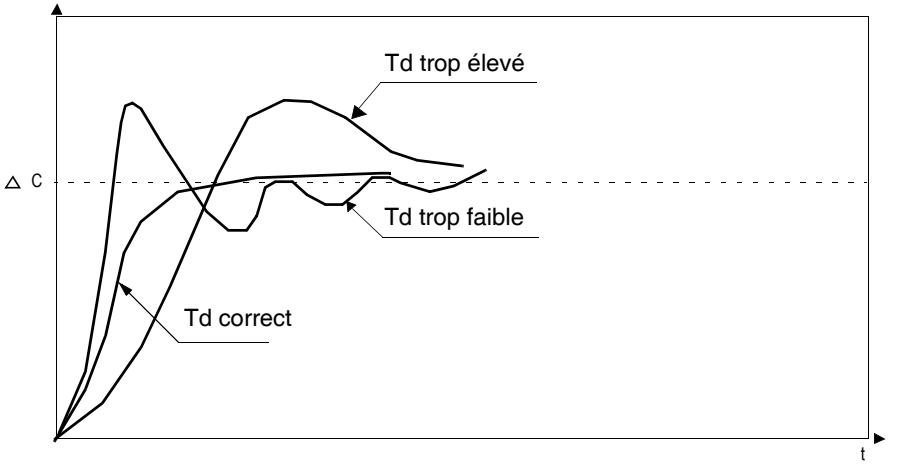


Note : Un Ti faible entraîne un niveau d'action intégrale élevé.

où Kp = gain proportionnel, Ti = temps d'intégration et Td = temps de dérivation.

Influence de l'action dérivée

L'action dérivée est anticipative. Dans la pratique, elle ajoute un terme qui prend en compte la vitesse de variation de l'écart, ce qui permet d'anticiper les modifications en accélérant les temps de réponse du processus lorsque l'écart augmente et en les ralentissant lorsqu'il diminue. Plus le niveau d'action dérivée est élevé (Td élevé), plus la réponse est rapide. Il convient de trouver un compromis approprié entre la vitesse et la stabilité. L'influence de l'action dérivée sur la réponse du processus à une division d'échelle est la suivante :



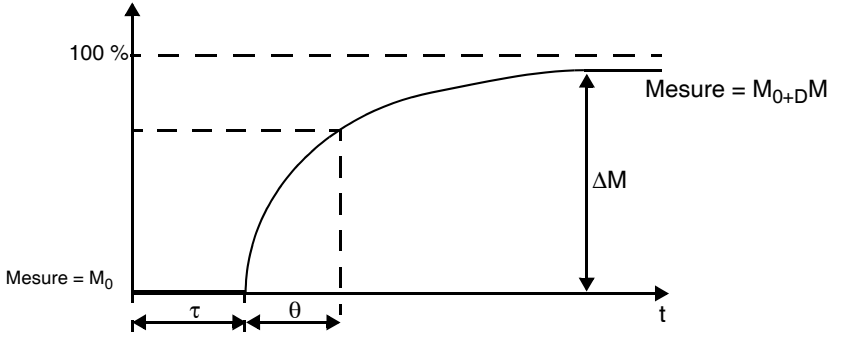
Limites de la boucle de contrôle du PID

Si le processus est assimilé à un premier ordre de temporisation pur avec une fonction de transfert : $(H(p)) = K\frac{(e^{(-\tau)p})}{(1+\theta p)}$

où :

τ = temporisation du modèle,

θ = constante de temps du modèle,



Les performances de contrôle du processus dépendent du rapport $\frac{\tau}{\theta}$

Le processus de contrôle du PID approprié est atteint dans le domaine suivant : 2-$\frac{\tau}{\theta}$-20

Pour $\frac{\tau}{\theta}$ <2, en d'autres termes pour des boucles de commande rapides (θ faible) ou pour des processus avec une temporisation importante (t élevé), le contrôle du processus PID n'est plus approprié. Dans ce cas, des algorithmes plus complexes doivent être utilisés.

Pour $\frac{\tau}{\theta}$ >20, un contrôle de processus utilisant un seuil plus l'hystérésis est suffisant.

Annexe 1 : Notions fondamentales de la théorie PID

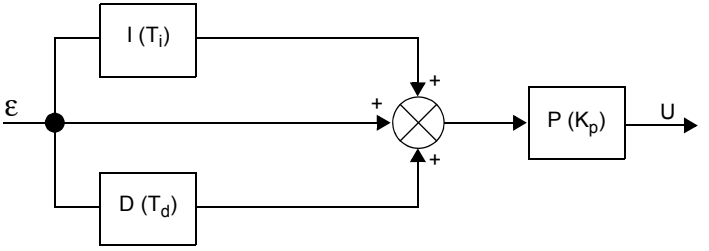
Introduction

La fonction du contrôle PID intégrée à tous les automates Twido permet de contrôler efficacement les processus industriels simples qui comprennent un déclencheur système (appelé Consigne dans ce document) et une propriété mesurable du système (appelé Mesure ou Variable de régulation).

Modèle de l'automate PID

L'automate PID Twido implémente une correction PID (série – parallèle) mixte (voir schéma du modèle PID ci-dessous) à partir d'une mesure et d'une consigne analogiques au format [0 – 10 000] et fournit une commande analogique au processus contrôlé analogique au même format.

La forme mixte du modèle de l'automate PID est décrite dans le schéma suivant :



Où

où :

- I = action **intégrale** (agissant indépendamment et parallèle à l'action dérivée),
- D = action **dérivée** (agissant indépendamment et parallèle à l'action intégrale),
- P = action **proportionnelle** (agissant en série sur la sortie associée des actions intégrales et dérivées,
- U = sortie de l'automate PID (alimentation ultérieure comme entrée dans le processus contrôlé.)

Loi de l'automate PID

L'automate PID comprend une association mixte (série - parallèle) du gain de l'automate (Kp) et des constantes de temps intégrales (Ti) et dérivées (Td). Ainsi, la loi de régulation PID utilisée par l'automate Twido a la forme suivante *(Eq.1) :*

$$u(i) = K_P \cdot \left\{ \varepsilon(i) + \frac{T_s}{T_i} \sum_{j=1}^{i} \varepsilon(j) + \frac{T_d}{T_s}[\varepsilon(i) - \varepsilon(i-1)] \right\}$$

Où

- Kp = Gain proportionnel de l'automate,
- Ti = Constante de temps intégrale,
- Td = Constante de temps dérivée,
- Ts = Période d'échantillonnage,
- $\varepsilon(i)$ = Ecart ($\varepsilon(i)$ = consigne – variable de régulation.)

Note : Deux algorithmes différents de calcul sont utilisés, selon la valeur de la constante de temps intégrale (Ti) :

- $Ti \neq 0$: Dans ce cas, un algorithme incrémentiel est utilisé.
- Ti = 0 : C'est le cas pour les processus de non intégration. Dans ce cas, un algorithme positionnel est utilisé, ainsi qu'un décalage +5 000 appliqué à la variable de sortie PID.

Pour plus de détails sur Kp, Ti et Td, reportez-vous au sous-chapitre *p. 596*.
Par déduction de *(equ.1)* et *(equ.1')*, le paramètre clé pour la régulation PID est la **période d'échantillonnage (Ts)**. La période d'échantillonnage dépend étroitement de la **constante de temps (τ)**, un paramètre intrinsèque au processus que le PID vise à contrôler. (Voir *p. 632*.)

Annexe 2 : Premier ordre avec modèle de temporisation

Présentation

Cette section présente le premier ordre avec le modèle de temporisation utilisé pour décrire différents processus industriels simples mais néanmoins importants, y compris les processus thermiques.

Premier ordre avec modèle de temporisation

On suppose que de simples processus thermiques (à un déclencheur) peuvent se rapprocher de manière adéquate d'un premier ordre avec un modèle de temporisation.

La fonction de transfert de ce processus boucle ouverte de premier ordre a la forme suivante dans le domaine Laplace *(equ.2) :*

$$\frac{S}{U} = \frac{k}{1 + \tau p} \cdot e^{-\theta p}$$

Où

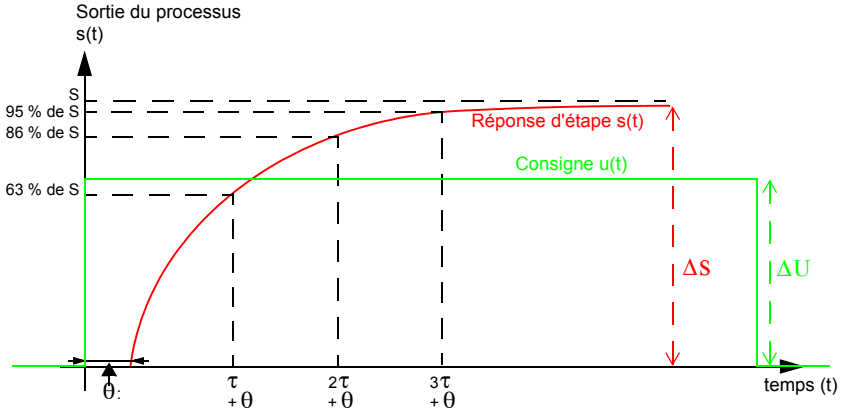
- k = gain statique,
- τ = constante de temps,
- θ = temps de retard,
- U = entrée du processus (sortie de l'automate PID),
- S = sortie du processus.

Constante de temps du processus τ

Le paramètre clé se la loi de réponse du processus *(equ.2)* est la **constante de temps τ.** Il s'agit d'un paramètre intrinsèque au processus à contrôler.

La constante de temps (τ) du système de premier ordre est définie par le temps (en secondes) mis par la variable de sortie du système pour atteindre 63 % de la sortie finale à partir du moment où le système commence à réagir au déclenchement de l'étape u(t).

Le schéma suivant illustre une réponse à processus de premier ordre type dans le cas d'un déclenchement d'étape :



Où

- k = gain statique calculé comme le ratio ΔS/ΔU,
- τ = temps pour une augmentation de 63 % = constante de temps,
- 2τ = temps pour une augmentation de 86 %,
- 3τ = temps pour une augmentation de 95 %.

Note : Lorsque l'auto tuning est implémenté, la période d'échantillonnage (Ts) doit être choisie dans la plage suivante : [τ/125 <Ts < τ/25]. Il est conseillé d'utiliser [Ts= τ/75]. (Voir *p. 615*.)

18.5 Instructions sur flottants

Présentation

Objet de ce sous-chapitre

Ce sous-chapitre décrit les instructions avancées sur flottants (voir *p. 33*) du langage TwidoSuite.

Les instructions de comparaisons et d'affectations sont décrites dans la section *p. 474*

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions arithmétiques sur flottant	635
Instructions trigonométriques	639
Instructions de conversion	641
Instructions de conversion Entier <-> Flottant	642

Instructions arithmétiques sur flottant

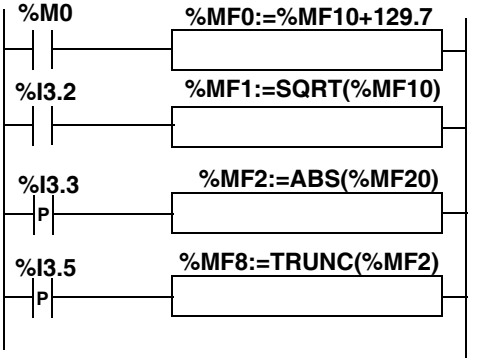
Généralités

Ces instructions permettent de réaliser une opération arithmétique entre deux opérandes ou sur un opérande.

+	Addition de deux opérandes	**SQRT**	Racine carrée d'un opérande
-	Soustraction de deux opérandes	**ABS**	Valeur absolue d'un opérande
*****	Multiplication de deux opérandes	**TRUNC**	Partie entière d'une valeur flottante
/	Division de deux opérandes	**EXP**	Exponentielle naturelle
LOG	Logarithme base 10	**EXPT**	Puissance d'un entier par un réel
LN	Logarithme népérien		

Structure

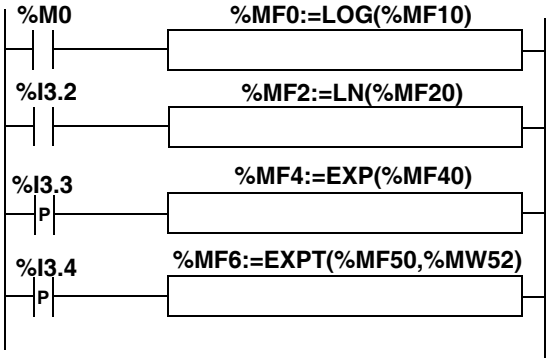
Langage schéma à contacts



Langage liste d'instructions

```
LD %M0
[%MF0:=%MF10+129.7]

LD %I3.2
[%MF1:=SQRT(%MF10)]

LDR %I3.3
[%MF2:=ABS(%MF20)]

LDR %I3.5
[%MF8:=TRUNC(%MF2)]
```

Langage schéma à contacts



Langage liste d'instructions

```
LD %M0
[%MF0:=LOG(%MF10]

LD %I3.2
[%MF2:=LN(%MF20)]

LDR %I3.3
[%MF4:=EXP(%MF40)]

LDR %I3.4
[%MF6:=EXPT(%MF50,%MW52)]
```

Syntaxe

Opérateurs et syntaxe des instructions arithmétiques sur flottant

Opérateurs	Syntaxe
+, - *, /	Op1:=Op2 Opérateur Op3
SQRT, ABS, TRUNC, LOG, EXP, LN	Op1:=Opérateur(Op2)
EXPT	Op1:=Opérateur (Op2,Op3)

Note : Lorsqu'on effectue une addition ou une soustraction entre 2 nombres flottants, les 2 opérandes doivent respecter la condition : $Op1 > Op2 \times 2^{-24}$, où Op1>Op2. Si cette condition n'est pas respectée, le résultat est égal à l'opérande 1 (Op1). Ce comportement est sans grande conséquence lorsqu'ils s'agit d'une opération isolée, puisque l'erreur résultante est très faible (2^{-24}), mais a des conséquences inattendues s'il s'agit d'un calcul itératif.
Exemple de l'instruction **%MF2:= %MF2 + %MF0** répétée indéfiniment. Si les conditions initiales sont %MF0 = 1.0 et %MF2= 0, on observe un blocage de la valeur de %MF2 à 16777216.
Il est donc déconseillé de programmer sans précaution des calculs itératifs. Si vous souhaitez néanmoins programmer ce type de calcul, il appartient à l'applicatif client de gérer les erreurs de troncature.

Opérandes des instructions arithmétiques sur flottant:

Opérateurs	Opérande 1 (Op1)	Opérande 2 (Op2)	Opérande 3 (Op3)
+, - *, /	%MFi	%MFi, %KFi, valeur immédiate	%MFi, %KFi, valeur immédiate
SQRT, ABS, LOG, EXP, LN	%MFi	%MFi, %KFi	[-]
TRUNC	%MFi, %MDi	%MFi, %KFi	[-]
EXPT	%MFi	%MFi, %KFi	%MWi, %KWi, valeur immédiate
Remarque : TwidoSuite empêche l'utilisation de la fonction avec %MWi comme Op1.			

Règles d'utilisation

- Les opérations sur flottants et sur entiers ne peuvent pas être mélangées directement. Les opérations de conversion (voir *p. 642*) assurent la conversion dans l'un ou l'autre de ces formats.
- Le bit système %S18 est géré de la même façon que pour les opérations sur entier (voir *p. 483*), le mot %SW17 (voir *p. 683*) indique la cause du défaut.
- Lorsque l'opérande de la fonction est un nombre non valide (par ex., un logarithme ou un nombre négatif), il produit un résultat infini ou indéterminé et modifie le bit %S18 en 1, le mot %SW17 indiquant la cause de l'erreur.

Note : Pour l'instruction TRUNC, le bit système %S17 n'est pas affecté.

Exemples d'instruction TRUNC avec %MDi

Le tableau ci-dessous fournit des exemples d'instruction TRUNC lorsque %MDi sert à stocker le résultat :

Exemple	Résultat
TRUNC (3.5)	3
TRUNC (324.18765)	324
TRUNC (927.8904)	927
TRUNC (-7.7)	-7
TRUNC (45.678E+20)	2 147 483 647 (mot double signé maximum) * %S18 est mis à 1
TRUNC (-94.56E+13)	- 2 147 483 648 (mot double signé minimum)* %S18 est mis à 1
***Remarque : Cet exemple s'applique à l'instruction TRUNC lorsqu'elle est utilisée avec %MDi. Lorsqu'elle est utilisée avec %MFi, l'instruction TRUNC ne présente pas de dépassement et ne dispose par conséquent pas de limite maximum/minimum.**	

Instructions trigonométriques

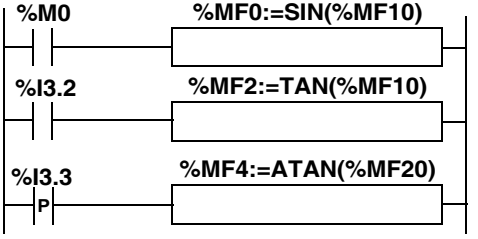
Généralités

Ces instructions permettent de réaliser des opérations trigonométriques.

SIN	sinus d'un angle exprimé en radian,	**ASIN**	arc sinus (résultat entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$)
COS	cosinus d'un angle exprimé en radian,	**ACOS**	arc cosinus (résultat entre 0 et π)
TAN	tangente d'un angle exprimée en radian.	**ATAN**	arc tangente (résultat entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$)

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %M0
[%MF0:=SIN(%MF10)]

LD %I3.2
[%MF2:=TAN(%MF10)]

LDR %I3.3
[%MF4:=ATAN(%MF20)]
```

Langage littéral structuré

```
IF %M0 THEN
 %MF0:=SIN(%MF10);
END_IF;
IF %I3.2 THEN
 %MF2:=TAN(%MF10);
END_IF;
IF %I3.3 THEN
 %MF4:=ATAN(%MF20);
END_IF;
```

Format Opérateurs, opérandes et format des instructions opérations trigonométriques:

Opérateurs	Format	Opérande 1 (Op1)	Opérande 2 (Op2)
SIN, COS, TAN, ASIN, ACOS, ATAN	Op1:=Opérateur(Op2)	%MFi	%MFi, %KFi

Règles d'utilisation

- lorsque l'opérande de la fonction est une valeur invalide (exemple : arc cosinus d'un nombre supérieur à 1), elle produit un résultat indéterminé ou infini et fait passer le bit %S18 à 1, le mot %SW17 (voir *p. 683*) indique la cause du défaut.
- les fonctions SIN/COS/TAN admettent en paramètre un angle entre -4096π et 4096π mais leur précision décroît progressivement pour les angles en dehors de l'intervalle -2π et $+2\pi$ en raison de l'imprécision apportée par le modulo 2π effectué sur le paramètre avant toute opération.

Instructions de conversion

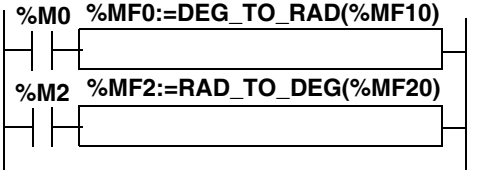
Généralités

Ces instructions permettent de réaliser des opérations de conversion.

DEG_TO_RAD	conversion de degré en radian, le résultat est la valeur de l'angle compris entre 0 et 2π
RAD_TO_DEG	conversion d'un angle exprimée en radian, le résultat est la valeur de l'angle compris entre 0 et 360 degrés

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %M0
[%MF0:=DEG_TO_RAD(%MF10)]

LD %M2
[%MF2:=RAD_TO_DEG(%MF20)]
```

Langage littéral structuré

```
IF %M0 THEN
 %MF0:=DEG_TO_RAD(%MF10);
END_IF;
IF %M2 THEN
 %MF2:=RAD_TO_DEG(%MF20);
END_IF;
```

Format

Opérateurs, opérandes et format des instructions de conversion:

Opérateurs	Format	Opérande 1 (Op1)	Opérande 2 (Op2)
DEG_TO_RAD RAD_TO_DEG	Op1:=Opérateur(Op2)	%MFi	%MFi, %KFi

Règles d'utilisation

L'angle à convertir doit être compris entre –737280,0 et +737280,0 (pour les conversions DEG_TO_RAD) ou entre -4096π et 4096π (pour les conversions RAD_TO_DEG).

Pour des valeurs non comprises entre ces bornes le résultat affiché sera + 1.#NAN, les bits %S18 et %SW17:X0 étant positionnés à 1.

Instructions de conversion Entier <-> Flottant

Généralités

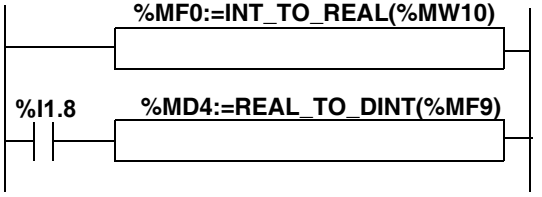
Quatre instructions de conversion sont proposées.

Liste des instructions de conversion entier<-> flottant :

INT_TO_REAL	conversion d'un mot entier --> flottant
DINT_TO_REAL	conversion d'un mot double (entier) --> flottant
REAL_TO_INT	conversion d'un flottant --> mot entier (le résultat est la valeur algébrique la plus proche)
REAL_TO_DINT	conversion d'un flottant --> mot double entier (le résultat est la valeur algébrique la plus proche)

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD TRUE
[%MF0:=INT_TO_REAL(%MW10)]

LD I1.8
[%MD4:=REAL_TO_DINT(%MF9)]
```

Langage littéral structuré

```
%MF0:=INT_TO_REAL(%MW10);
IF %I1.8 THEN
 %MD4:=REAL_TO_DINT(%MF9);
END_IF;
```

Format

Opérateurs et format (conversion d'un mot entier --> flottant) :

Opérateurs	Format
INT_TO_REAL	Op1=INT_TO_REAL(Op2)

Opérandes (conversion d'un mot entier --> flottant) :

Opérande 1 (Op1)	Opérande 2 (Op2)
%MFi	%MWi,%KWi

Exemple : conversion mot entier --> flottant : 147 --> 1,47e+02

Opérateurs et format (conversion mot double entier --> flottant) :

Opérateurs	Format
DINT_TO_REAL	Op1=DINT_TO_REAL(Op2)

Opérandes (conversion mot double entier --> flottant) :

Opérande 1 (Op1)	Opérande 2 (Op2)
%MFi	%MDi,%KDi

Exemple : conversion mot double entier --> flottant : 68905000 --> 6,8905e+07

Opérateurs et format (conversion flottant --> mot entier ou mot double entier) :

Opérateurs	Format
REAL_TO_INT	Op1=Opérateur(Op2)
REAL_TO_DINT	

Opérandes (conversion flottant --> mot entier ou mot double entier) :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi	%MFi, %KFi
Mots doubles	%MDi	%MFi, %KFi

Exemple :

conversion flottant --> mot entier : 5978.6 --> 5979

conversion flottant --> mot double entier : -1235978.6 --> -1235979

Note : Si, lors d'une conversion réel vers entier (ou réel vers entier mot double), la valeur flottante est en dehors des bornes du mot (ou du mot double), le bit %S18 est positionné à 1.

Précision d'arrondi

La norme IEEE 754 définit 4 modes d'arrondi pour les opérations sur flottant.

Le mode utilisé par les instructions ci-dessus est le mode "arrondi au plus près":

"si les valeurs représentables les plus proches sont à égale distance du résultat théorique, la valeur fournie sera celle dont le bit de poids faible est égal à 0".

Dans certains cas, le résultat de l'arrondi peut donc prendre une valeur par défaut ou une valeur par excès.

Par exemple :

Arrondi de la valeur 10,5 -> 10

Arrondi de la valeur 11,5 -> 12

18.6 Instructions ASCII

Présentation

Objet de cette section

Cette sous-section décrit les instructions ASCII avancées du langage TwidoSuite.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instruction ROUND	646
Conversion d'une valeur ASCII en valeur entière	648
Conversion d'une valeur entière en valeur ASCII	650
Conversion d'une valeur ASCII en flottant	652
Conversion d'un flottant en valeur ASCII	654

Instruction ROUND

Description de l'instruction

L'instruction ROUND arrondit une représentation flottante stockée dans une chaîne ASCII.

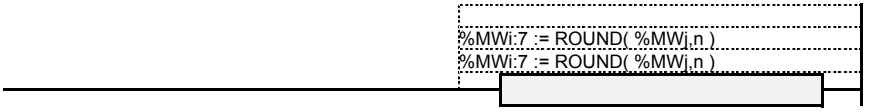
Syntaxe de l'instruction

Pour une instruction ROUND, utilisez la syntaxe suivante :
Op1 := ROUND(Op2,Op3).

Par exemple,
En liste d'instructions :

%MWi:7 := ROUND(%MWj,n)

En langage schéma à contacts :



Paramètres de l'instruction

Le tableau ci-dessous décrit les paramètres de la fonction ROUND :

ASCII	Description
Op1	%MW qui stocke le résultat
Op2	%MW qui contient le flottant à arrondir
Op3	Nombre de chiffres significatifs requis pour arrondir Entier compris entre 1 et 8

Règles de l'instruction

Les règles de l'instruction ROUND sont les suivantes :

- L'opérande est toujours arrondi à la valeur inférieure.
- Le délimiteur de la chaîne d'opérandes sert à délimiter la chaîne résultante.
- Il peut s'agir de tout caractère ASCII non compris dans l'intervalle ['0' - '9'] ([16#30 - 16#39]), sauf pour les caractères suivants :
  - point '.' (16#2E)
  - signe moins '-' (16#2D)
  - signe plus '+' (16#2B)
  - exp 'e' ou 'E' (16#65 ou 16#45)
- Le résultat et l'opérande ne doivent pas dépasser 13 octets : la taille maximum d'une chaîne ASCII est de 13 octets.

Erreurs de syntaxe

TwidoSuite vérifie la syntaxe. Les exemples suivants risquent d'entraîner des erreurs de syntaxe :

Syntaxe incorrecte	Syntaxe correcte
%MW10:= ROUND (%MW1,4) ":7" manquant dans le résultat	%MW10:7 := ROUND (%MW1,4)
%MW10:13:= ROUND (%MW1,4) %MW10:n où n ≠ 7 est incorrect	%MW10:7 := ROUND (%MW1,4)

Exemples

Le tableau ci-dessous fournit des exemples d'instruction ROUND :

Exemple	Résultat
ROUND ("12.3456e+6", 4)	"12.34e+6"
ROUND ("987654321", 5)	"987650000"
ROUND ("-11.1", 8)	"-11.1"
ROUND ("NAN")	"NAN"

Conversion d'une valeur ASCII en valeur entière

Description de l'instruction

L'instruction de conversion d'une valeur ASCII en valeur entière convertit une chaîne ASCII en une valeur entière.

Syntaxe de l'instruction

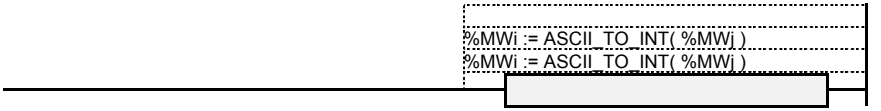
Pour l'instruction de conversion d'une valeur ASCII en valeur entière, utilisez la syntaxe suivante : **Op1 := ASCII_TO_INT(Op2)**.

Par exemple,

En liste d'instructions :

%MWi := ASCII_TO_INT(%MWj)

En langage schéma à contacts :



Paramètres de l'instruction

Le tableau suivant décrit les paramètres de la fonction de conversion d'une valeur ASCII en valeur entière :

ASCII	Description
Op1	%MW qui stocke le résultat
Op2	%MW ou %KW

Règles de conversion

Les règles de conversion d'une valeur ASCII en valeur entière sont les suivantes :

- La fonction lit toujours l'octet de poids fort en premier.
- Tout caractère ASCII non compris dans l'intervalle ['0' - '9'] ([16#30 - 16#39]) est considéré comme un caractère de fin, à l'exception du signe moins '-' (16#2D) lorsqu'il est placé comme premier caractère.
- En cas de dépassement (>32767 ou <-32768), le bit système %S18 (erreur ou dépassement arithmétique) est mis à 1 et la valeur 32767 ou -32768 est renvoyée.
- Si le premier caractère de l'opérande est un caractère de "fin", la valeur 0 est renvoyée et le bit %S18 est mis à 1.

Exemples

Soient les données ASCII suivantes stockées dans les mots %MW10 à %MW13 :

Paramètre	Valeur hexadécimale	Valeur ASCII
%MW10	16#3932	'9', '2'
%MW11	16#3133	'1', '3'
%MW12	6#2038	' ', '8'
%MW13	16#3820	'8', ' '

Le tableau ci-dessous fournit des exemples de conversion de valeur ASCII en valeur entière :

Exemple	Résultat
%MW20 := ASCII_TO_INT(%MW10)	%MW20 = 29318
%MW20 := ASCII_TO_INT(%MW12)	%MW20 = 8
%MW20 := ASCII_TO_INT(%MW13)	%MW20 = 0 et %S18 est mis à 1

Conversion d'une valeur entière en valeur ASCII

Description de l'instruction

L'instruction de conversion d'une valeur entière en valeur ASCII convertit une valeur entière en valeur de chaîne ASCII.

Syntaxe de l'instruction

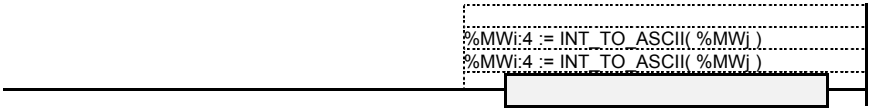
Pour l'instruction de conversion d'une valeur entière en valeur ASCII, utilisez la syntaxe suivante : **Op1 := INT_TO_ASCII(Op2)**.

Par exemple,

En liste d'instructions :

%MWi:4 := INT_TO_ASCII(%MWj)

En langage schéma à contacts :



Paramètres de l'instruction

Le tableau suivant décrit les paramètres de la fonction de conversion d'une valeur entière en valeur ASCII :

ASCII	Description
Op1	%MW qui stocke le résultat
Op2	%MW, %KW, %SW, %IW, %QW ou tout MOT (Les valeurs immédiates ne sont pas acceptées.)

Règles de conversion

Les règles de conversion d'une valeur entière en valeur ASCII sont les suivantes :

- La fonction écrit toujours l'octet de poids fort en premier.
- Le caractère de fin est "Espace" (ASCII 32).
- La fonction détermine automatiquement le nombre de %MW devant contenir des valeurs ASCII (de 1 à 4).

Erreurs de syntaxe

TwidoSuite vérifie la syntaxe. Les exemples suivants risquent d'entraîner des erreurs de syntaxe :

Syntaxe incorrecte	Syntaxe correcte
%MW10 := INT_TO_ASCII (%MW1) ":4" manquant dans le résultat	%MW10:4 := INT_TO_ASCII (%MW1)
%MW10:n := INT_TO_ASCII (%MW1) %MW10:n où n ≠ 4 est incorrect	%MW10:4 := INT_TO_ASCII (%MW1)

Exemples

Pour l'instruction MW10:4 := INT_TO_ASCII(%MW1) :

Si...	**Alors...**	
Valeur entière	**Valeur hexadécimale**	**Valeur ASCII**
%MW1 = 123	%MW10 = 16#3231	'2', '1'
	%MW11 = 16#2033	' ', '3'
%MW1 = 45	%MW10 = 16#3534	'5', '4'
	%MW11 = 16#0020	' '
%MW1 = 7	%MW10 = 16#2037	' ', '7'
%MW1 = 12369	%MW10 = 16#3145	'1', '-'
	%MW11 = 16#3332	'3', '2'
	%MW10 = 16#3936	'9', '6'
	%MW11 = 16#0020	' '

Conversion d'une valeur ASCII en flottant

Description de l'instruction

L'instruction de conversion d'une valeur ASCII en flottant convertit une chaîne ASCII en valeur flottante.

Syntaxe de l'instruction

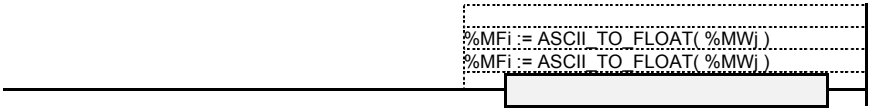
Pour l'instruction de conversion d'une valeur ASCII en flottant, utilisez la syntaxe suivante : **Op1 := ASCII_TO_FLOAT(Op2)**.

Par exemple,

En liste d'instructions :

%MFi := ASCII_TO_FLOAT(%MWj)

En langage schéma à contacts :



Paramètres de l'instruction

Le tableau suivant décrit les paramètres de la fonction de conversion d'une valeur ASCII en flottant :

ASCII	Description
Op1	%MF
Op2	%MW ou %KW

Règles de conversion

Les règles de conversion d'une valeur ASCII en flottant sont les suivantes :

- La fonction lit toujours l'octet de poids fort en premier.
- Tout caractère ASCII non compris dans l'intervalle ['0' - '9'] ([16#30 - 16#39]) est considéré comme un caractère de "fin", sauf les caractères suivants :
  - point '.' (16#2E)
  - signe moins '-' (16#2D)
  - signe plus '+' (16#2B)
  - exp 'e' ou 'E' (16#65 ou 16#45)
- Le format de la chaîne ASCII peut être une notation scientifique (par ex., "-2.34567e+13") ou décimale (par ex., "9826.3457").
- En cas de dépassement (si le résultat du calcul est >3.402824E+38 ou <-3.402824E+38) :
  - le bit système %S18 (erreur ou dépassement arithmétique) est mis à 1 ;
  - %SW17:X3 est mis à 1 ;
  - la valeur +/- 1.#INF (valeur infinie + ou -) est renvoyée.
- Si le résultat du calcul est compris entre -1,175494E-38 et 1,175494E-38, il est arrondi à 0,0.
- Si l'opérande n'est pas un nombre :
  - la valeur 1.#NAN est renvoyée ;
  - le bit %SW17:X0 est mis à 1.

Exemples

Soient les données ASCII suivantes stockées dans les mots %MW10 à %MW14 :

Paramètre	Valeur hexadécimale	Valeur ASCII
%MW10	16#382D	'8', '-'
%MW11	16#322E	'2', '.'
%MW12	16#3536	'5', '6'
%MW13	16#2B65	'+', 'e'
%MW14	16#2032	' ','2'

Le tableau ci-dessous fournit des exemples de conversion de valeur ASCII en valeur flottante :

Exemple	Résultat
%MW20 := ASCII_TO_FLOAT(%MW10)	%MF20 = -826.5
%MW20 := ASCII_TO_FLOAT(%MW11)	%MF20 = 1.#QNAN
%MW20 := ASCII_TO_FLOAT(%MW12)	%MF20 = 6500.0
%MW20 := ASCII_TO_FLOAT(%MW13)	%MF20 = 1.#QNAN
%MW20 := ASCII_TO_FLOAT(%MW14)	%MF20 = 2.0

Conversion d'un flottant en valeur ASCII

Description de l'instruction

L'instruction de conversion d'un flottant en valeur ASCII convertit une valeur flottante en valeur de chaîne ASCII.

Syntaxe de l'instruction

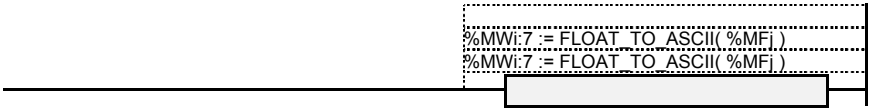
Pour l'instruction de conversion d'un flottant en valeur ASCII, utilisez la syntaxe suivante : **Op1 := FLOAT_TO_ASCII(Op2)**.

Par exemple,

En liste d'instructions :

```
%MWi:7 := FLOAT_TO_ASCII( %MFj )
```

En langage schéma à contacts :



Paramètres de conversion

Le tableau suivant décrit les paramètres de la fonction de conversion d'un flottant en valeur ASCII :

ASCII	Description
Op1	%MW
Op2	%MF ou %KF

Règles de conversion

Les règles de conversion d'un flottant en valeur ASCII sont les suivantes :

- La fonction écrit toujours l'octet de poids fort en premier.
- La représentation s'effectue à l'aide d'une notation scientifique conventionnelle.
- Le résultat "Infini" ou "Pas un nombre" renvoie la chaîne "NAN".
- Le caractère de fin est "Espace" (ASCII 32).
- La fonction détermine automatiquement le nombre de %MW devant contenir des valeurs ASCII.
- La précision de conversion est de 6 chiffres.

Erreurs de syntaxe

TwidoSuite vérifie la syntaxe. Les exemples suivants risquent d'entraîner des erreurs de syntaxe :

Syntaxe incorrecte	Syntaxe correcte
%MW10 := FLOAT_TO_ASCII (%MF1) ":7" manquant dans le résultat	%MW10:7 := FLOAT_TO_ASCII(%MF1)
%MW10:n := FLOAT_TO_ASCII (%MF1) %MW10:n où n ≠ 7 est incorrect	%MW10:7 := FLOAT_TO_ASCII(%MF1)

Exemples

Pour l'instruction %MW10:7 := FLOAT_TO_ASCII(%MF1) :

Nombre à convertir	Résultat
1 234 567 800	1,23456e+09
0,000000921	9,21e-07
9,87654321	9,87654
1 234	1,234e+03

18.7 Instructions sur tableaux d'objets

Présentation

Objet de ce sous-chapitre

Ce sous chapitre décrit les instructions spécifiques aux tableaux :

- de doubles mots,
- de flottants.

Les instructions d'affectation sur tableaux sont décrites dans le chapitre des "instructions élémentaires" (voir *p. 479*).

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonction de sommation sur tableaux	657
Fonction de comparaison de tableaux	659
Fonctions de recherche sur tableaux	661
Fonctions de recherche de valeurs maximum et minimum sur tableaux	663
Nombre d'occurrences d'une valeur dans un tableau	664
Fonction décalage circulaire sur un tableau	665
Fonction de tri sur tableau	667
Fonction d'interpolation sur tableau de flottants	668
Fonction de moyenne des valeurs d'un tableau de flottants	672

Fonction de sommation sur tableaux

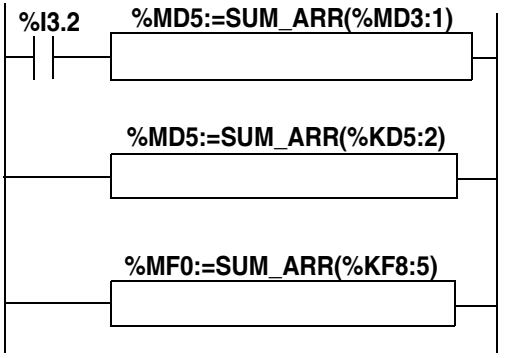
Généralités

La fonction SUM_ARR effectue la somme de tous les éléments d'un tableau d'objet :

- si le tableau est constitué de mots doubles, le résultat est donné sous la forme d'un mot double
- si le tableau est constitué de mots flottants, le résultat est donné sous la forme d'un mot flottant

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[%MD5:=SUM_ARR(%MD3:1)]
%MD5:=SUM_ARR(%KD5:2)
%MF0:=SUM_ARR(%KF8:5)
```

Format

Format de l'instruction de sommation sur tableau :

```
Res:=SUM_ARR(Tab)
```

Paramètres de l'instruction de sommation sur tableau

Type	Résultat (res)	Tableau (Tab)
Tableaux de mots doubles	%MDi	%MDi:L,%KDi:L
Tableaux de flottants	%MFi	%MFi:L,%KFi:L

Note : le bit %S18 est mis à 1 lorsque le résultat n'est pas dans les bornes du format double mot suivant l'opérande tableau.

Exemple

```
%MD4:=SUM(%MD30:4)
```

où %MD30=10, %MD32=20, %MD34=30, %MD36=40

%MD4:=10+20+30+40

Fonction de comparaison de tableaux

Généralités

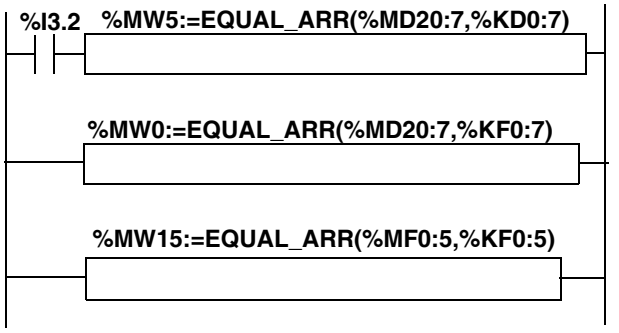
La fonction EQUAL _ARR effectue la comparaison de 2 tableaux élément par élément.

Si une différence apparaît, le rang des premiers éléments dissemblables est retourné sous forme d'un mot, sinon la valeur retournée est égale à -1.

La comparaison est effectuée sur la totalité du tableau.

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[%MW5:=EQUAL_ARR(%MD20:7,KD0:7)]
```

Langage littéral structuré

```
%MW0:=EQUAL_ARR(%MD20:7,%KF0:7)

%MW15:=EQUAL_ARR(%MF0:5,%KF0:5)
```

Format

Format de l'instruction de comparaison de tableaux :

```
Res:=EQUAL_ARR(Tab1,Tab2)
```

Paramètres des instructions de comparaison de tableaux :

Type	Résultat (Res)	Tableaux (Tab1 et Tab2)
Tableaux de mots doubles	%MWi	%MDi:L,%KDi:L
Tableaux de flottants	%MWi	%MFi:L,%KFi:L

Note :
- les tableaux doivent être obligatoirement de même longueur et de même type.

Exemple

```
%MW5:=EQUAL_ARR(%MD30:4,%KD0:4)
```

Comparaison des 2 tableaux :

Rang	Tableau de Mots	Tableaux de Constantes	Différence
0	%MD30=10	%KD0=10	=
1	%MD32=20	%KD2=20	=
2	%MD34=30	%KD4=60	Différent
3	%MD36=40	%KD6=40	=

Le mot %MW5 vaut 2 (premier rang différent).

Fonctions de recherche sur tableaux

Généralités

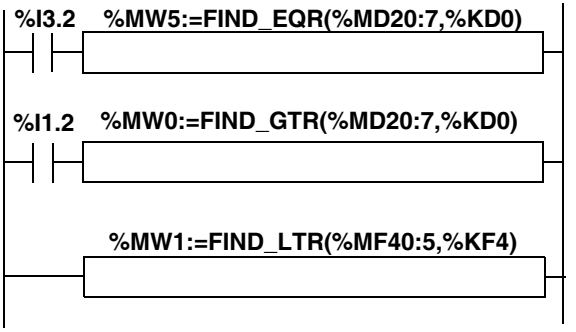
3 fonctions de recherche sont proposées :

- **FIND_EQR** : recherche de la position dans un tableau de mots doubles ou de flottants du premier élément égal à une valeur donnée
- **FIND_GTR** : recherche de la position dans un tableau de mots doubles ou de flottants du premier élément supérieur à une valeur donnée
- **FIND_LTR** : recherche de la position dans un tableau de mots doubles ou de flottants du premier élément inférieur à une valeur donnée

Le résultat de ces instructions est égal au rang du premier élément trouvé ou à -1 si la recherche est infructueuse.

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[%MW5:=FIND_EQR(%MD20:7,KD0)]
LD %I1.2
[%MW0:=FIND_GTR(%MD20:7,%KD0)]
%MW1:=FIND_LTR(%MF40:5,%KF4)
```

Format

Format des instructions de recherche sur tableaux :

Fonction	Format
FIND_EQR	Res:=Fonction(Tab,Val)
FIND_GTR	
FIND_LTR	

Paramètres des instructions de recherche sur tableaux de flottants et mots doubles :

Type	Résultat (Res)	Tableau (Tab)	Valeur (val)
Tableaux de flottants	%MWi	%MFi:L,%KFi:L	%MFi,%KFi
Tableaux de mots doubles	%MWi	%MDi:L,%KDi:L	%MDi,%KDi

Exemple

```
%MW5:=FIND_EQR(%MD30:4,%KD0)
```

Recherche de la position du premier mot double =%KD0=30 dans le tableau :

Rang	Tableau de Mots	Résultat
0	%MD30=10	-
1	%MD32=20	-
2	%MD34=30	%MW5=2 (valeur du rang)
3	%MD36=40	-

Fonctions de recherche de valeurs maximum et minimum sur tableaux

Généralités

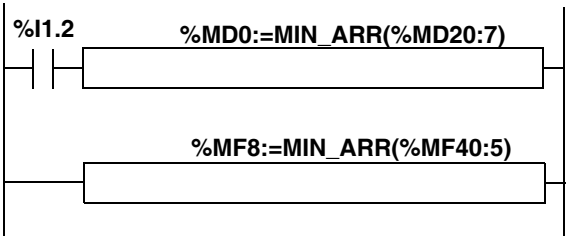
2 fonctions de recherche sont proposées :

- **MAX_ARR** : recherche de la valeur maximum dans un tableau de mots doubles et de flottants
- **MIN_ARR** : recherche de la valeur minimum dans un tableau de mots doubles et de flottants

Le résultat de ces instructions est égal à la valeur maximum (ou minimum) trouvée dans le tableau.

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I1.2
[%MD0:=MIN_ARR(%MD20:7)]
%MF8:=MIN_ARR(%MF40:5)
```

Format

Format des instructions de recherche de valeurs maxi et mini sur tableaux :

Fonction	Format
MAX_ARR	Res:=Fonction(Tab)
MIN_ARR	

Paramètres des instructions de recherche de valeurs maxi et mini sur tableaux:

Type	Résultat (Res)	Tableau (Tab)
Tableaux de mots doubles	%MDi	%MDi:L,%KDi:L
Tableaux de flottants	%MFi	%MFi:L,%KFi:L

Nombre d'occurrences d'une valeur dans un tableau

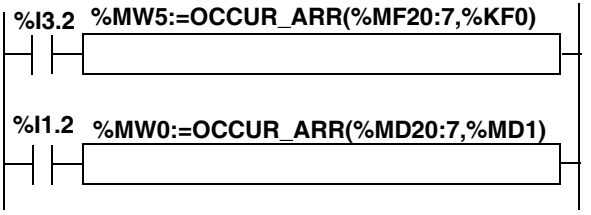
Généralités

La fonction de recherche proposée :

- **OCCUR_ARR** : effectue la recherche dans un tableau de mots doubles ou de flottants du nombre d'éléments égaux à une valeur donnée

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[%MW5:=OCCUR_ARR(%MF20:7,%KF0)]
LD %I1.2
[%MW0:=OCCUR_ARR(%MD20:7,%MD1)
```

Format

Format des instructions de recherche de valeurs maxi et mini sur tableaux :

Fonction	Format
OCCUR_ARR	Res:=Fonction(Tab,Val)

Paramètres des instructions de recherche de valeurs maxi et mini sur tableaux :

Type	Résultat (Res)	Tableau (Tab)	Valeur (Val)
Tableaux de mots doubles	%MWi	%MDi:L,%KDi:L	%MDi,%KDi
Tableaux de flottants	%MFi	%MFi:L,%KFi:L	%MFi,%KFi

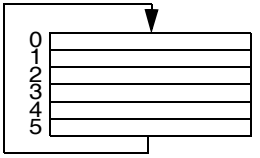
Fonction décalage circulaire sur un tableau

Généralités

2 fonctions de décalage sont proposées :

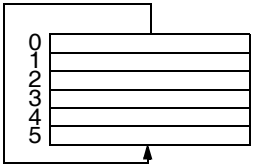
- **ROL_ARR** : réalise le décalage circulaire de n positions de haut en bas des éléments du tableau de flottants

Illustration des fonctions ROL_ARR



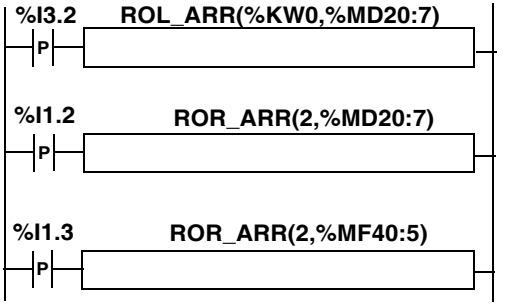
- **ROR_ARR** : réalise le décalage circulaire de n positions de bas en haut des éléments du tableau de flottants.

Illustration des fonctions ROR_ARR



Structure

Langage schéma à contacts



Langage liste d'instructions

```
LDR %I3.2
[ROL_ARR(%KW0,%MD20:7)]
LDR %I1.2
 [ROR_ARR(2,%MD20:7)]
LDR %I1.3
[ROR_ARR(2,%MF40:5)]
```

Format

Syntaxe des instructions de décalage circulaire sur tableaux de doubles mots ou de flottants **ROL_ARR** et **ROR_ARR**

Fonction	Format
ROL_ARR	Fonction(n,Tab)
ROR_ARR	

Paramètres des instructions de décalage circulaire sur tableaux de flottants: **ROL_ARR** et **ROR_ARR** :

Type	Nombre de positions (n)	Tableau (Tab)
Tableaux de flottants	%MWi, valeur immédiate	%MFi:L
Tableaux de mots doubles	%MWi, valeur immédiate	%MDi:L

Note : si la valeur de n est négative ou nulle, aucun décalage n'est effectué.

Fonction de tri sur tableau

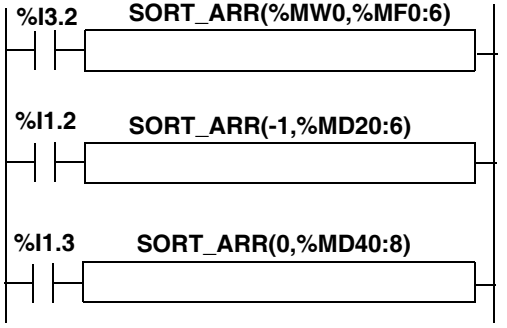
Généralités

La fonction de tri proposée est la suivante :

- **SORT_ARR** : réalise les tris par ordre croissant ou décroissant des éléments d'un tableau de mots doubles ou de flottants et range ce qui en résulte dans ce même tableau.

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[SORT_ARR(%MW20,%MF0:6)]
LD %I1.2
[SORT_ARR(-1,%MD20:6)]
LD %I1.3
[SORT_ARR(0,%MF40:8)
```

Format

Format des fonctions de tri sur tableaux :

Fonction	Format
SORT_ARR	Fonction(sens,Tab)

- le paramètre "sens" donne l'ordre du tri : sens > 0 le tri se fait par ordre croissant, sens < 0 le tri s'effectue par ordre décroissant, sens = 0 aucun tri n'est effectué.
- le résultat (tableau trié) est retourné dans le paramètre Tab (tableau à trier).

Paramètres des fonctions de tri sur tableaux :

Type	Sens du tri	Tableau (Tab)
Tableaux de mots doubles	%MWi, valeur immédiate	%MDi:L
Tableaux de flottants	%MWi, valeur immédiate	%MFi:L

Fonction d'interpolation sur tableau de flottants

Vue d'ensemble

La fonction **LKUP** sert à interpoler un ensemble de données flottantes X par rapport à Y pour une valeur X donnée.

Règle d'interpolation

La fonction LKUP suit la règle d'interpolation linéaire, comme défini dans l'équation suivante :

(équation 1 :) $$Y = Y_i + \left[\frac{(Y_{i+1} - Y_i)}{(X_{i+1} - X_i)} \cdot (X - X_i)\right]$$

pour $X_i \le X \le X_{i+1}$, où $i = 1 \ldots (m-1)$;

si les valeurs X_i sont classées par ordre croissant : $X_1 \le X_2 \le \ldots X \ldots \le X_{m-1} \le X_m$.

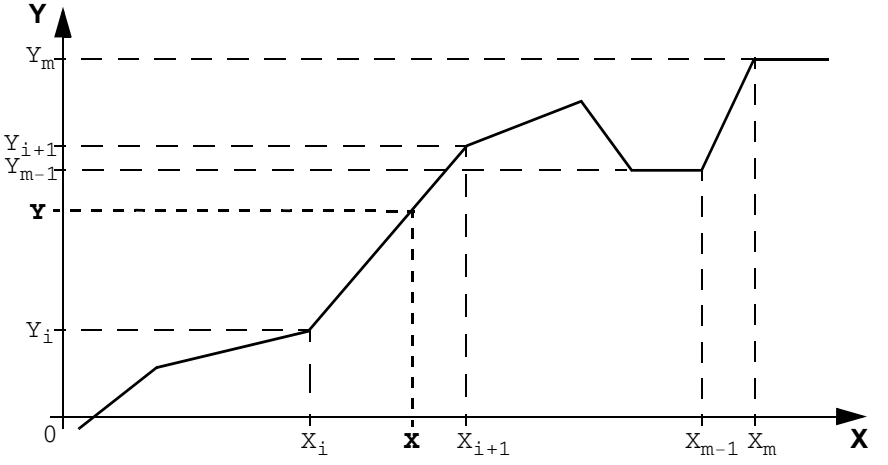
Note : Si les deux valeurs Xi consécutives sont égales ($X_i=X_{i+1}=X$), l'équation (1) fournit une exception invalide. Dans ce cas, pour faire face à cette exception, l'algorithme suivant est utilisé à la place de l'équation (1) :

(équation 2 :) $$Y = \left[\frac{(Y_{i+1} - Y_i)}{2}\right]$$

pour $X_i = X_{i+1} = X$, où $i = 1 \ldots (m-1)$.

Représentation graphique de la règle d'interpolation linéaire

Le graphique suivant illustre la règle d'interpolation linéaire décrite ci-dessus :



Format de la fonction LKUP

La fonction LKUP utilise trois opérandes, dont deux sont des attributs de fonction, comme décrit dans le tableau suivant :

Format	Opérande 1 (Op1) Variable de sortie	Opérande 2 (Op2) Valeur (X) définie par l'utilisateur	Opérande 3 (Op3) Tableau de variables (X_i,Y_i) défini par l'utilisateur
[Op1: = LKUP(Op2,Op3)]	%MWi	%MF0	Entier, %MWi ou %KWi

Définition de Op1

Op1 est le mot mémoire qui contient la variable de sortie de la fonction d'interpolation.

Selon la valeur d'Op1, l'utilisateur peut savoir si l'interpolation a fonctionné ou échoué, ainsi que les causes de l'échec, comme mentionné dans le tableau suivant :

Op1 (%MWi)	Description
0	Interpolation réussie
1	Erreur d'interpolation : Tableau incorrect, $X_m < X_{m-1}$
2	Erreur d'interpolation : Op2 hors plage, $X < X_1$
4	Erreur d'interpolation : Op2 hors plage, $X > X_m$
8	Taille du tableau des données incorrecte : • Op3 est défini comme un nombre impair ou • Op3 < 6.

Note : Op1 **ne** contient pas la valeur d'interpolation calculée (Y). Pour une valeur (X) donnée, le résultat de l'interpolation (Y) est contenu dans %MF2 du tableau Op3 (voir *Définition de Op3* ci-dessous).

Définition de Op2

Op2 est la variable flottant (%MF0 du tableau flottant Op3) qui contient la valeur (X) définie par l'utilisateur qui permet de calculer la valeur (Y) interpolée :

- La plage valide pour Op2 est la suivante : $X_1 \leq Op2 \leq X_m$.

Définition de Op3

Op3 définit la taille (Op3 / 2) du tableau flottant où les paires de données (X_i,Y_i) sont enregistrées.

Les données X_i et Y_i sont enregistrées dans des objets flottants avec des index pairs, commençant à %MF4 (notez que les objets flottants %MF0 et %MF2 sont réservés respectivement à la consigne X de l'utilisateur et à la valeur Y interpolée).

Avec un tableau de (m) paires de données (X_i,Y_i), l'index supérieur (u) du tableau flottant (%MFu) est défini en utilisant les relations suivantes :

- *(équation 3 :)* $Op3 = 2 \cdot m$;
- *(équation 4 :)* $u = 2 \cdot (Op3 - 1)$.

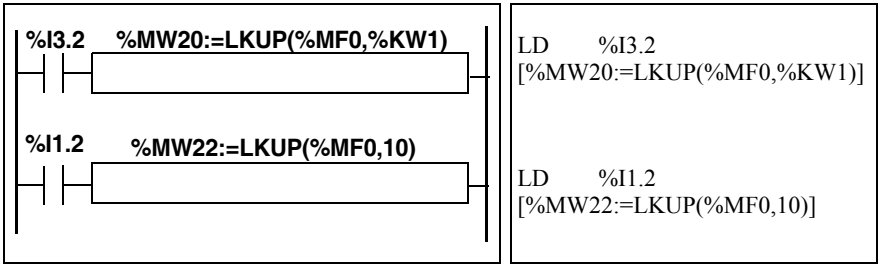
La structure du tableau flottant Op3 (%MFi) est similaire à celle de l'exemple suivant (où Op3=8) :

(X)		(X_1)		(X_2)		(X_3)	
%MF0		**%MF4**		**%MF8**		**%MF12**	
	%MF2		**%MF6**		**%MF10**		**%MF14**
	(Y)		(Y_1)		(Y_2)		(Y_3)
							(Op3=8)

Note : En raison de la structure du tableau flottant ci-dessus, Op3 doit respecter les exigences suivantes. Sinon, cela déclenche une erreur de la fonction LKUP :
- Op3 est un chiffre pair et
- Op3 ≥ 6 (au moins 2 points de données doivent être disponibles pour permettre une interpolation linéaire).

Structure

Les opérations d'interpolation sont effectuées de la façon suivante :



Exemple

L'exemple suivant illustre l'utilisation d'une fonction d'interpolation LKUP :

```
[%MW20:=LKUP(%MF0,10)]
```

Dans cet exemple :

- %MW20 est Op1 (la variable de sortie).
- %MF0 est la valeur (X) définie par l'utilisateur dont la valeur (Y) correspondante doit être calculée par interpolation linéaire.
- %MF2 enregistre la valeur calculée (Y) générée par l'interpolation linéaire.
- 10 est Op3 (comme indiqué par l'*équation 3* ci-dessus). Il définit la taille du tableau flottant. Elément de classement le plus élevé %MFu, où =18 est indiqué par l'*équation 4,* ci-dessus.

Quatre paires de points de données sont stockées dans le tableau Op3 [%MF4..%MF18] :

- %MF4 contient X_1,%MF6 contient Y_1.
- %MF8 contient X_2,%MF10 contient Y_2.
- %MF12 contient X_3,%MF14 contient Y_3.
- %MF16 contient X_4,%MF18 contient Y_4.

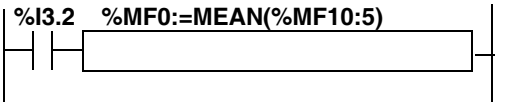
Fonction de moyenne des valeurs d'un tableau de flottants

Généralités

La fonction **MEAN** permet de calculer la moyenne des valeurs d'un nombre donné de point d'un tableau de flottants.

Structure

Langage schéma à contacts



Langage liste d'instructions

```
LD %I3.2
[%MF0:=MEAN(%MF10:5)]
```

Format

Format de la fonction de calcul de moyenne d'un tableau de flottants :

Fonction	Format
MEAN	Result=Fonction(Op1)

Paramètres de la fonction de calcul d'un nombre donné L de valeurs d'un tableau de flottants :

Opérande (Op1)	Résultat (Result)
%MFi:L, %KFi:L	%MFi

Bits système et mots système

19

Présentation

Objet de ce chapitre

Ce chapitre offre une présentation des bits système et des mots systèmes pouvant être utilisés lors de la création des programmes de régulation d'automates Twido.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Bits système (%S)	674
Mots système (%SW)	683

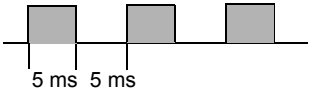
Bits système (%S)

Introduction

Cette section présente des informations détaillées sur la fonction des bits système, ainsi que sur leur mode de régulation.

Description détaillée

Le tableau suivant présente une vue d'ensemble des bits système, ainsi que leur mode de régulation :

Bit système	Fonction	Description	Etat initial	Contrôle
%S0	Démarrage à froid	Normalement à l'état 0, ce bit est mis à 1 par : ● une reprise de l'alimentation avec perte de données (défaillance de la pile) ; ● le programme utilisateur ou l'éditeur de tables d'animation ; ● l'afficheur. Ce bit est mis à 1 au cours de la première scrutation. Il est ensuite remis à zéro par le système avant la scrutation suivante.	0	S ou U->S
%S1	Démarrage à chaud	Normalement à l'état 0, ce bit est mis à 1 par : ● une reprise de l'alimentation avec sauvegarde des données ; ● le programme utilisateur ou l'éditeur de tables d'animation ; ● l'afficheur. Il est ensuite remis à zéro par le système une fois la scrutation terminée.	0	S ou U->S
%S4 %S5 %S6 %S7	Base temps : 10 ms Base temps : 100 ms Base temps : 1 s Base temps : 1 min	Les changements d'état de ces bits sont cadencés par une horloge interne. Ils ne sont pas synchronisés avec la scrutation de l'automate. Exemple : %S4  	-	S
%S8	Test du câblage	Initialement à l'état 1, ce bit est utilisé pour le test du câblage lorsque l'automate est à l'état "non configuré". Pour modifier la valeur de ce bit, utilisez les touches de l'afficheur pour changer l'état des sorties souhaitées : ● à l'état 1, mise à zéro des sorties ; ● à l'état 0, test du câblage autorisé.	1	U

Bit système	Fonction	Description	Etat initial	Contrôle
%S9	Mise à zéro des sorties	Normalement à l'état 0, ce bit peut être mis à 1 par le programme ou par le terminal (dans l'éditeur de tables d'animations) : • à l'état 1, la valeur des sorties est forcée sur 0 lorsque l'automate est en mode d'exécution (RUN) ; • à l'état 0, les sorties sont mises à jour normalement.	0	U
%S10	Défaillance d'E/S	Normalement à l'état 1, ce bit peut être mis à 0 par le système lorsqu'une défaillance d'E/S est détectée.	1	S
%S11	Débordement du chien de garde	Normalement à l'état 0, ce bit peut être mis à 1 par le système lorsque la durée d'exécution du programme (durée de scrutation) dépasse la durée de scrutation maximale (chien de garde logiciel). Le débordement du chien de garde fait passer l'automate en mode suspendu (HALT).	0	S
%S12	Automate en mode d'exécution (RUN)	Ce bit reflète l'état d'exécution de l'automate. Le système règle le bit sur 1 lorsque l'automate est en cours d'exécution. A l'arrêt, lors de l'initialisation du système et pour tout autre état, ce bit est mis à 0.	0	S
%S13	Premier cycle en mode d'exécution (RUN)	Normalement à l'état 0, ce bit est mis à 1 par le système au cours de la première scrutation une fois l'automate passé en mode d'exécution (RUN).	1	S
%S17	Dernier bit éjecté	Normalement réglé sur 0. Il est défini par le système en fonction de la valeur du dernier bit éjecté. Il indique la valeur du dernier bit éjecté.	0	S->U
%S18	Débordement ou erreur arithmétique	Normalement à l'état 0, ce bit est mis à 1 en cas de débordement découlant de l'exécution d'une opération sur 16 bits générant : • un résultat supérieur à +32 767 ou inférieur à -32 768, en simple longueur ; • un résultat supérieur à +2 147 483 647 ou inférieur à -2 147 483 648, en double longueur ; • un résultat supérieur à +3,402824E+38 ou inférieur à -3,402824E+38, en flottant ; • une division par 0 ; • la racine carrée d'un nombre négatif ; • une conversion BTI ou ITB non significative : valeur BCD hors plage. Doit être testé par le programme utilisateur après chaque opération pouvant provoquer un débordement, puis remis à zéro par l'utilisateur en cas de débordement.	0	S->U

Bit système	Fonction	Description	Etat initial	Contrôle
%S19	Débordement de la période de scrutation (scrutation périodique)	Normalement à l'état 0, ce bit est mis à 1 par le système en cas de débordement d'une période de scrutation (durée de scrutation supérieure à la durée définie par l'utilisateur au moment de la configuration ou programmée dans %SW0). Ce bit est remis à zéro par l'utilisateur.	0	S->U
%S20	Débordement d'index	Normalement à l'état 0, ce bit est mis à 1 lorsque le repère de l'objet indexé devient inférieur à 0 ou supérieur à sa taille maximale. Doit être testé par le programme utilisateur après chaque opération pouvant provoquer un débordement, puis remis à zéro en cas de débordement.	0	S->U
%S21	Initialisation du GRAFCET	Normalement à l'état 0, ce bit est mis à 1 par : ● une reprise à froid, %S0=1 ; ● le programme utilisateur, uniquement dans la section du programme de prétraitement, à l'aide de l'instruction SET (S %S21) ou d'une bobine SET -(S)- %S21 ; ● le terminal. A l'état 1, il provoque l'initialisation du GRAFCET. Tous les pas actifs sont désactivés et les pas initiaux sont activés. Il est ensuite remis à zéro par le système après l'initialisation du GRAFCET.	0	U->S
%S22	Remise à zéro du GRAFCET	Normalement à l'état 0, ce bit ne peut être mis à 1 par le programme qu'au cours du prétraitement. A l'état 1, il provoque la désactivation des pas de l'ensemble du GRAFCET. Il est remis à zéro par le système au début de l'exécution du traitement séquentiel.	0	U->S
%S23	Prépositionnement et gel du GRAFCET	Normalement à l'état 0, ce bit ne peut être mis à 1 par le programme que dans le module du programme de prétraitement. A l'état 1, il valide le prépositionnement du GRAFCET. Le maintien de ce bit sur la valeur 1 a pour effet de geler le GRAFCET (gel du graphique). Il est remis à zéro par le système au début de l'exécution du traitement séquentiel pour garantir l'évolution du GRAFCET à partir de la situation de gel.	0	U->S

Bit système	Fonction	Description	Etat initial	Contrôle
%S24 (1)	Afficheur	Normalement à l'état 0, ce bit peut être réglé sur 1 par l'utilisateur : ● à l'état 0, l'afficheur fonctionne normalement ; ● à l'état 1, l'afficheur est gelé, conserve l'affichage courant, le clignotement est désactivé et les touches ne sont plus prises en compte. (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	U->S
%S25 (1)	Choix d'un mode d'affichage sur l'afficheur	Vous pouvez choisir entre deux modes d'affichage sur l'afficheur à 2 lignes : mode de données et mode normal. ● Si %S25 = 0, le mode normal est activé. Sur la première ligne, vous pouvez écrire un nom d'objet (un mot système, un mot mémoire, un bit système, etc.). Sur la deuxième ligne, vous pouvez lire sa valeur. ● Si %S25 = 1, le mode de données est activé. Sur la première ligne, vous pouvez afficher la valeur %SW68. Sur la deuxième ligne, vous pouvez afficher la valeur %SW69. Si %S25 = 1, le clavier de l'opérateur est désactivé. **Remarque :** La version du microprogramme doit être supérieure à 3.0. (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	U
%S26 (1)	Choix d'une valeur avec signe ou sans signe sur l'afficheur	Vous pouvez choisir entre deux types de valeurs : avec signe ou sans signe. ● Si %S26 = 0, l'affichage de la valeur avec signe (-32768 à 32767) est activé. Les signes +/- apparaissent à chaque début de ligne. ● Si %S26 = 1, l'affichage de la valeur sans signe (0 à 65535) est activé. %S26 peut être utilisé uniquement si %S25 = 1. **Remarque :** La version du microprogramme doit être supérieure à 3.0. (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	U

Bit système	Fonction	Description	Etat initial	Contrôle
%S31	Masque d'événement	Normalement réglé sur 1 ; ● à l'état 0, les événements ne peuvent pas être exécutés et sont mis en attente ; ● à l'état 1, les événements peuvent être exécutés. Ce bit peut être réglé sur son état initial 1 par l'utilisateur et le système (lors d'un redémarrage à froid).	1	U->S
%S38	Permission des événements à être placés dans la file d'événements	Normalement réglé sur 1 ; ● à l'état 0, les événements ne peuvent pas être placés dans la file d'événements ; ● à l'état 1, les événements sont placés dans la file d'événements dès qu'ils sont détectés. Ce bit peut être réglé sur son état initial 1 par l'utilisateur et le système (lors d'un redémarrage à froid).	1	U->S
%S39	Saturation de la file d'événements	Normalement réglé sur 0 : ● à l'état 0, tous les événements sont reportés ; ● à l'état 1, au moins un événement est perdu. Ce bit peut être réglé sur 0 par l'utilisateur et le système (lors d'un redémarrage à froid).	0	U->S
%S50	Mise à jour de la date et de l'heure à l'aide des mots %SW49 à %SW53	Normalement à l'état 0, ce bit peut être mis à 1 ou 0 par le programme ou l'afficheur : ● à l'état 0, la date et l'heure peuvent être lues ; ● à l'état 1, la date et l'heure peuvent être mises à jour. L'horodateur interne de l'automate est mis à jour sur un front descendant de %S50.	0	U->S
%S51	Etat de l'horloge calendaire	Normalement à l'état 0, ce bit peut être mis à 1 ou 0 par le programme ou l'afficheur : ● à l'état 0, la date et l'heure sont cohérentes ; ● à l'état 1, la date et l'heure doivent être initialisées par l'utilisateur. Lorsque ce bit est réglé sur 1, les données de l'horloge calendaire ne sont pas valides. Il est possible que la date et l'heure n'aient jamais été configurées, que le niveau de la pile soit faible ou que la constante de correction de l'automate ne soit pas valide (jamais configurée, différence entre la valeur d'horloge corrigée et valeur enregistrée ou la valeur hors plage). Le passage de l'état 1 à 0 force l'écriture de la constante de correction sur l'horodateur.	0	U->S

Bit système	Fonction	Description	Etat initial	Contrôle
%S52	Erreur de l'horodateur	Ce bit géré par le système indique que la correction de l'horodateur n'a pas été saisie et que la date et l'heure sont erronées : ● à l'état 0, la date et l'heure sont cohérentes ; ● à l'état 1, la date et l'heure doivent être initialisées.	0	S
%S59	Mise à jour de la date et de l'heure à l'aide du mot %SW59	Normalement à l'état 0, ce bit peut être mis à 1 ou 0 par le programme ou l'afficheur : ● à l'état 0, le mot système %SW59 n'est pas géré ; ● à l'état 1, la date et l'heure sont incrémentées ou décrémentées en fonction des fronts montants sur les bits de contrôle réglés dans %SW59.	0	U
%S66 (1)	Affichage du voyant BAT activé/désactivé (uniquement sur les automates qui prennent en charge une pile externe : automates TWDLC••40DRF)	Ce bit système peut être défini par l'utilisateur. Permet d'allumer/éteindre le voyant BAT : ● à l'état 0, le voyant BAT est allumé (il est remis à zéro par le système lors de la mise sous tension) ; ● à l'état 1, le voyant BAT est éteint (le voyant reste éteint même si l'alimentation de la pile externe est faible ou si aucune pile n'est présente dans le compartiment à pile). (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	S ou U->S
%S69 (1)	Affichage du voyant STAT utilisateur	A l'état 0, le voyant STAT est éteint. A l'état 1, le voyant STAT est allumé. (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	U
%S75 (1)	Etat de la pile externe (uniquement sur les automates qui prennent en charge une pile externe : automates TWDLC••40DRF)	Ce bit système est défini par le système. Il indique l'état de la pile externe et peut être lu par l'utilisateur : ● à l'état 0, la pile externe fonctionne normalement ; ● à l'état 1, l'alimentation de la pile est faible ou la pile ne se trouve pas dans le compartiment à pile. (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.	0	S
%S95	Restauration des mots mémoire	Ce bit peut être défini lorsque les mots mémoire ont été préalablement enregistrés dans la mémoire EEPROM interne. Le système remet ensuite ce bit à 0 et le nombre de mots mémoire restaurés est défini dans %SW97.	0	U

Bit système	Fonction	Description	Etat initial	Contrôle
%S96	Programme de sauvegarde OK	Ce bit peut être lu à n'importe quel moment (soit par le programme ou lors d'un réglage), en particulier après un démarrage à froid ou un redémarrage à chaud : ● à l'état 0 si l'automate contient une application non valide ; ● à l'état 1 si l'automate contient une application valide.	0	S
%S97	Enregistrement %MW OK	Ce bit peut être lu à n'importe quel moment (soit par le programme ou lors d'un réglage), en particulier après un démarrage à froid ou un redémarrage à chaud : ● à l'état 0, l'enregistrement %MW est incorrect ; ● à l'état 1, l'enregistrement %MW est correct.	0	S
%S100	Raccordement du câble de communication TwidoSuite	Indique si le câble de communication TwidoSuite est raccordé : ● à l'état 1, soit le câble de communication TwidoSuite n'est pas raccordé, soit TwidoSuite est connecté. ● à l'état 0, le câble de liaison distante TwidoSuite est raccordé.	-	S
%S101	Changement de repère d'un port (protocole Modbus)	Permet de changer le repère d'un port en utilisant les mots système %SW101 (port 1) et %SW102 (port 2). Pour cela, il faut mettre %S101 à l'état 1. ● A l'état 0, il est impossible de changer le repère. La valeur de %SW101 et %SW102 correspond au repère actuel du port. ● A l'état 1, il est possible de changer le repère en modifiant les valeurs de %SW101 (port 1) et %SW102 (port 2). Après avoir modifié les valeurs des mots système, il faut remettre %S101 à l'état 0. **Remarque :** ● En mode connecté, il est impossible de modifier l'adresse du port 2 à l'aide du bit système %S101 et du mot système %SW102. ● %SW102 et le port 2 ne sont pas disponibles pour l'automate TWDLEDCK1 Twido Extreme.	0	U

Bit système	Fonction	Description	Etat initial	Contrôle
%S103 %S104 (1)	Utilisation du protocole ASCII	Permet d'utiliser le protocole ASCII sur le port Comm 1 (%S103) ou Comm 2 (%S104). Le protocole ASCII sera configuré à l'aide des mots système %SW103 et %SW105 pour le port Comm 1 et %SW104 et %SW106 pour le port Comm 2. ● A l'état 0, le protocole utilisé est celui configuré dans TwidoSuite. ● A l'état 1, le protocole ASCII est utilisé sur le port Comm 1 (%S103) ou Comm 2 (%S104). Dans ce cas, il faut configurer auparavant les mots système %SW103 et %SW105 pour le port Comm 1 et %SW104 et %SW106 pour le port Comm 2. **Remarque :** %S104, %SW104, %SW106 et Comm 2 ne sont pas disponibles pour l'automate Twido Extreme TWDLEDCK1.	0	U
%S110	Echanges de liaison distante	Ce bit est remis à zéro par le programme ou par le terminal : ● à l'état 1 pour un maître, tous les échanges de liaison distante (E/S distantes uniquement) sont terminés ; ● à l'état 1 pour un esclave, l'échange avec le maître est terminé.	0	S->U
%S111	Echange de liaison distante unique	● A l'état 0 pour un maître, un échange de liaison distante unique est terminé. ● A l'état 1 pour un maître, un échange de liaison distante unique est actif.	0	S
%S112	Connexion de liaison distante	● A l'état 0 pour un maître, la liaison distante est activée. ● A l'état 1 pour un maître, la liaison distante est désactivée.	0	U
%S113	Configuration/ fonctionnement de la liaison distante	● A l'état 0 pour un maître ou un esclave, la configuration/le fonctionnement de la liaison distante est correct(e). ● A l'état 1 pour un maître, la configuration ou le fonctionnement de la liaison distante présente une erreur. ● A l'état 1 pour un esclave, la configuration ou le fonctionnement de la liaison distante présente une erreur.	0	S->U

Bit système	Fonction	Description	Etat initial	Contrôle
%S118	Erreur d'E/S distantes	Normalement à l'état 1, ce bit peut être mis à 0 lorsqu'une défaillance d'E/S est détectée sur la liaison distante.	1	S
%S119	Erreur d'E/S locales	Normalement à l'état 1, ce bit peut être mis à 0 lorsqu'une défaillance d'E/S est détectée sur la base automate. %SW118 détermine la nature de la défaillance. Remis à 1 lorsque la défaillance est résolue.	1	S
%S120	Dépassement entrée PWM0 (%IW0.7)	Normalement réglé sur 0. Ce bit peut être défini sur 1 par l'automate lorsque la fréquence du signal à %IW0.7 se trouve hors de la plage valide pour %IW0.7. Remise sur 0 par l'utilisateur.	0	S->U
%S121	Dépassement entrée PWM1 (%IW0.8)	Normalement réglé sur 0. Ce bit peut être défini sur 1 par l'automate lorsque la fréquence du signal à %IW0.7 se trouve hors de la plage valide pour %IW0.8. Remise sur 0 par l'utilisateur.	0	S->U

Note : (1) Ce bit système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.

Description des abréviations utilisées dans le tableau précédent

Tableau des abréviations :

Abréviation	Description
S	Contrôlé par le système
U	Contrôlé par l'utilisateur
U->S	Réglé sur 1 par l'utilisateur, remis à 0 par le système
S->U	Mis à 1 par le système, réglé sur 0 par l'utilisateur

Mots système (%SW)

Introduction Cette section présente des informations détaillées sur la fonction des mots système, ainsi que sur leur mode de régulation.

Description détaillée Le tableau suivant fournit des informations détaillées sur la fonction des mots système, ainsi que sur leur mode de régulation :

Mots système	Fonction	Description	Contrôle
%SW0	Période de scrutation de l'automate (tâche périodique)	Modifie la période de scrutation de l'automate, définie lors de la configuration à l'aide du programme utilisateur dans l'éditeur de tables d'animation.	U
%SW1	Enregistrement de la valeur d'un événement périodique	Modifie le temps de cycle [5 à 255 ms] d'un événement périodique, sans perdre la valeur de la période enregistrée dans la zone Evénement périodique de la fenêtre Mode de scrutation. Vous permet de récupérer la valeur de la période enregistrée dans la zone Evénement périodique : ● lors d'un démarrage à froid ou ● si la valeur que vous écrivez dans %SW1 se trouve en dehors de la plage [5-255]. La valeur %SW1 peut être modifiée à chaque fin de cycle, dans le programme ou dans la table d'animation, sans avoir besoin d'arrêter le programme. Les temps de cycle peuvent être observés correctement lorsque le programme est en cours d'exécution.	U
%SW6	Etat de l'automate	Etat de l'automate : 0 = NO CONFIG (Non configuré) 2 = STOP (Arrêté) 3 = RUN (Exécution) 4 = HALT (Suspendu)	S

Mots système	Fonction	Description	Contrôle

%SW7	Etat de l'automate	• Bit [0] : sauvegarde/restauration en cours :
		  • à l'état 1 si une sauvegarde/restauration est en cours ;
		  • à l'état 0 si la sauvegarde/restauration est terminée ou non active.
		• Bit [1] : configuration de l'automate correcte :
		  • à l'état 1 si la configuration est correcte.
		• Bit [3..2] : bits d'état EEPROM :
		  • 00 = Pas de cartouche
		  • 01 = Cartouche EEPROM 32 Ko *
		  • 10 = Cartouche EEPROM 64 Ko *
		  • 11 = Réservé pour une utilisation ultérieure
		• Bit [4] : application dans RAM différente de l'EEPROM :
		  • à l'état 1 si l'application RAM est différente de l'EEPROM.
		• Bit [5] : application RAM différente de la cartouche* :
		  • à l'état 1 si l'application RAM est différente de la cartouche.
		• Bit [6] non utilisé (état 0)
		• Bit [7] : automate réservé :
		  • à l'état 1 si réservé.
		• Bit [8] : application en mode écriture :
		  • à l'état 1 si l'application est protégée.
		• Bit [9] non utilisé (état 0)
		• Bit [10] : port série 2 installé* :
		  • à l'état 1 si installé.
		• Bit [11] : type de port série 2* : (0 = EIA RS-232, 1 = EIA RS-485) :
		  • à l'état 0 = EIA RS-232
		  • à l'état 1 = EIA RS-485
		• Bit [12] : application valide en mémoire interne :
		  • à l'état 1 si l'application est valide.
		• Bit [13] : application valide en cartouche* :
		  • à l'état 1 si l'application est valide.
		• Bit [14] : application valide en RAM :
		  • à l'état 1 si l'application est valide.
		• Bit [15] : prêt pour exécution :
		  • à l'état 1 si prêt pour l'exécution.
		
		Remarque : * La cartouche EEPROM et le port série 2 ne sont pas disponibles pour l'automate TWDLEDCK1 Twido Extreme.
			S

%SW11	Valeur du chien de garde logiciel (Watchdog)	Contient la valeur maximale du chien de garde (10 à 500 ms) définie par la configuration.	U

Mots système	Fonction	Description	Contrôle
%SW14	Version commerciale, Vxx.yy	Par exemple, si % SW14 = 0232 : ● 8 MSB = 02 en hexadécimal, puis xx = 2 en décimal ● 8 LSB = 32 en hexadécimal, puis yy = 50 en décimal Par conséquent, la version commerciale est 2.50. **Remarque :** La version du microprogramme doit être supérieure ou égale à 2.5.	S
%SW15	Correctif du microprogramme, Pzz	Par exemple, si % SW15 = 0005 : ● 8 MSB n'est pas utilisé ● 8 LSB = 05 en hexadécimal, puis zz = 5 en décimal Par conséquent, le correctif du microprogramme est P05. **Remarque :** La version du microprogramme doit être supérieure ou égale à 2.5.	S
%SW16	Version du microprogramme, Vxx.yy	Par exemple, si % SW16 = 0232 : ● 8 MSB = 02 en hexadécimal, puis xx = 2 en décimal ● 8 LSB = 32 en hexadécimal, puis yy = 50 en décimal Par conséquent, la version du microprogramme est 2.50. **Remarque :** La version du microprogramme doit être supérieure ou égale à 2.5.	S
%SW17	Etat par défaut pour opération flottante	Lorsqu'une défaillance est détectée dans une opération arithmétique flottante, le bit %S18 est à l'état 1 et le statut par défaut de %SW17 est mis à jour selon le codage suivant : ● Bit [0] : opération incorrecte, le résultat n'est pas un nombre (1.#NAN ou -1.#NAN). ● Bit 1 : réservé. ● Bit 2 : division par 0, le résultat est infini (-1.#INF ou 1.#INF). ● Bit 3 : résultat en valeur absolue supérieur à +3,402824e+38, le résultat est infini (-1.#INF ou 1.#INF).	S et U
%SW18-%SW19	Compteur de temporisation absolu 100 ms	Le compteur utilise deux mots : ● **%SW18** représente le mot de poids faible. ● **%SW19** représente le mot de poids fort.	S et U
%SW20 à %SW27	Fournit l'état des modules esclaves CANopen dotés d'une adresse de nœud comprise entre 1 et 16.	Pour plus d'informations, reportez-vous à la section *p. 289*.	S
%SW30	Durée de la dernière scrutation	Affiche la durée d'exécution du dernier cycle de scrutation de l'automate (en ms). **Remarque :** Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.	S

<table>
<tr><th>Mots système</th><th>Fonction</th><th colspan="3">Description</th><th>Contrôle</th></tr>
<tr><td>%SW31</td><td>Durée de scrutation maximale</td><td colspan="3">Affiche la durée d'exécution du plus long cycle de scrutation de l'automate (en ms), depuis le dernier démarrage à froid.
Remarques :
● Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.
● Pour garantir une détection correcte d'un signal des impulsions lorsque l'option d'entrée à mémorisation d'état est sélectionnée, la largeur d'impulsion (T_{ON}) et la période (P) doivent répondre aux deux exigences suivantes :
● $T_{ON} \geq 1$ ms
● La période du signal d'entrée (P) doit suivre la règle d'échantillonnage de Nyquist-Shannon qui déclare que la période du signal d'entrée doit correspondre à au moins deux fois le temps de scrutation maximal (%SW31) :
$P \geq 2$ x %SW31.
Remarque : Si cette condition n'est pas remplie, il risque de manquer certaines impulsions.</td><td>S</td></tr>
<tr><td>%SW32</td><td>Durée de scrutation minimale</td><td colspan="3">Affiche la durée d'exécution du cycle de scrutation de l'automate le plus court (en ms), depuis le dernier démarrage à froid.
Remarque : Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.</td><td>S</td></tr>
<tr><td></td><td rowspan="9">Informations d'état des objets Entrée/sortie PGN</td><td>Numéro de l'objet PGN</td><td></td><td>Contenu</td><td rowspan="9">U</td></tr>
<tr><td>%SW33</td><td>3-2</td><td>1-0</td><td rowspan="8">4 bits par PGN :
0 = Etat normal
1 = PGN reçu sans erreur
2 = force l'écriture sur la sortie PGN
4 = erreur PGN (entrée ou sortie)</td></tr>
<tr><td>%SW34</td><td>7-6</td><td>5-4</td></tr>
<tr><td>%SW35</td><td>11-10</td><td>9-8</td></tr>
<tr><td>%SW36</td><td>15-14</td><td>13-12</td></tr>
<tr><td>%SW37</td><td>19-18</td><td>17-16</td></tr>
<tr><td>%SW38</td><td>23-22</td><td>21-20</td></tr>
<tr><td>%SW39</td><td>27-26</td><td>25-24</td></tr>
<tr><td>%SW40</td><td>31-30</td><td>29-28</td></tr>
<tr><td>%SW48</td><td>Nombre d'événements</td><td colspan="3">Affiche le nombre d'événements exécutés depuis le dernier démarrage à froid. (Compte tous les événements à l'exception des événements périodiques.)
Remarque : A l'état 0 (après chargement de l'application et démarrage à froid). Cette valeur s'incrémente à chaque exécution d'un événement.</td><td>S</td></tr>
</table>

Mots système	Fonction	Description		Contrôle
%SW49 %SW50 %SW51 %SW52 %SW53	Horodateur	Fonctions RTC : mots contenant les valeurs courantes de date et d'heure (en BCD) :		S et U
		%SW49	xN jour de la semaine (N=1 pour lundi)	
		%SW50	00SS secondes	
		%SW51	HHMM heure et minute	
		%SW52	MMJJ mois et jour	
		%SW53	SSAA siècle et année	
		Ces mots sont contrôlés par le système lorsque le bit **%S50** est réglé sur 0. Ces mots peuvent être écrits par le programme utilisateur ou par le terminal, lorsque le bit **%S50** est paramétré sur 1. Sur un front descendant de **%S50**, l'horodateur interne de l'automate est mis à jour à partir des valeurs écrites dans ces mots.		
%SW54 %SW55 %SW56 %SW57	Date et heure du dernier arrêt	Mots système contenant la date et l'heure de la dernière coupure secteur ou du dernier arrêt de l'automate (en BCD) :		S
		%SW54	SS secondes	
		%SW55	HHMM heure et minute	
		%SW56	MMJJ mois et jour	
		%SW57	SSAA siècle et année	
%SW58	Code du dernier arrêt	Affiche le code indiquant la cause du dernier arrêt :		S
		1 =	Front de l'entrée Run/Stop	
		2 =	Arrêt en cas de défaillance logicielle (dépassement de la scrutation de l'automate)	
		3 =	Commande d'arrêt (Stop)	
		4 =	Coupure secteur	
		5 =	Arrêt en cas de défaillance matérielle	

Mot système	Fonction	Description			Contrôle
%SW59	Réglage de la date courante	Règle la date courante. Contient deux jeux de 8 bits permettant de régler la date courante. L'opération est toujours effectuée sur le front montant du bit. Ce mot est activé par le bit **%S59**.			U
		Incrément	**Décrément**	**Paramètre**	
		bit 0	bit 8	Jour de la semaine	
		bit 1	bit 9	Secondes	
		bit 2	bit 10	Minutes	
		bit 3	bit 11	Heures	
		bit 4	bit 12	Jours	
		bit 5	bit 13	Mois	
		bit 6	bit 14	Années	
		bit 7	bit 15	Siècles	
%SW60	Correction RTC	Valeur de correction de l'horodateur			U
%SW63	Code d'erreur du bloc EXCH1	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - timeout écoulé 6 - émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage de réception plus important que la table de réception 13 - interruption du traitement EXCH par l'automate			S
%SW64 (1)	Code d'erreur du bloc EXCH2	Code d'erreur EXCH2 : Voir %SW63. (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.			S

Mot système	Fonction	Description	Contrôle
%SW65 (1)	Code d'erreur du bloc EXCH3	Le code d'erreur EXCH3 est uniquement implémenté sur les automates Twido TWDLCAE40DRF prenant en charge Ethernet. 1-4, 6-13: voir %SW63. (Remarque : Le code d'erreur 5 est incorrect. Il est remplacé par les codes d'erreur Ethernet 109 et 122, qui sont décrits ci-dessous.) Les codes d'erreur suivants sont spécifiques à Ethernet : 101 - aucune adresse IP de ce type 102 - la connexion TCP est interrompue 103 - aucun socket disponible (toutes les voies de connexion sont occupées) 104 - le réseau ne fonctionne pas 105 - le réseau est inaccessible 106 - le réseau a interrompu la connexion lors de la réinitialisation 107 - la connexion a été abandonnée par le poste 108 - la connexion a été réinitialisée par le poste 109 - timeout écoulé pour la connexion 110 - rejet de la tentative de connexion 111 - l'hôte ne fonctionne pas 120 - index inconnu (le périphérique distant n'est pas indexé dans le tableau de configuration) 121 - erreur fatale (MAC, puce, adresse IP double) 122 - timeout de réception écoulé après l'envoi des données 123 - initialisation d'Ethernet en cours (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.	S
%SW67	Fonction et type d'automate	Contient les informations suivantes : ● Bit de type d'automate [0 -11] ● 8B0 = TWDLC•A10DRF ● 8B1 = TWDLC•A16DRF ● 8B2 = TWDLMDA20DUK/DTK ● 8B3 = TWDLC•A24DRF ● 8B2 = TWDLMDA20DUK/DTK ● 8B6 = TWDLMDA20DRT ● 8B8 = TWDLC•A40DRF ● 8B9 = TWDLC•E40DRF ● 8BA = TWDLEDCK1 ● Bits 12, 13, 14 et 15 non utilisés = 0	S

Mots système	Fonction	Description	Contrôle
%SW68 (1) et %SW69 (1)	Eléments à afficher simultanément sur l'afficheur à 2 lignes	Si %S25 = 1, le mode d'affichage de données est activé. Le clavier de l'opérateur est désactivé. %SW68 et %SW69 peuvent être affichés sur l'afficheur à 2 lignes : ● la valeur %SW68 sur la première ligne, ● la valeur %SW69 sur la deuxième ligne. **Remarque :** La version du microprogramme doit être supérieure ou égale à 3.0. (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.	U
%SW73 (1) et %SW74 (1)	Etat du système AS-Interface	● Bit [0] : à l'état 1 si la configuration est correcte. ● Bit [1] : à l'état 1 si l'échange de données est activé. ● Bit [2] : à l'état 1 si le module est en mode local. ● Bit [3] : à l'état 1 si l'instruction ASI_CMD est terminée. ● Bit [4] : à l'état 1 si erreur de l'instruction ASI_CMD en cours. (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.	S et U
%SW76 à %SW79	Décompteurs 1-4	Ces quatre mots sont utilisés comme temporisateurs à 1 ms. Ils sont décrémentés de manière individuelle par le système, toutes les millisecondes, si leur valeur est positive. Cela donne quatre décompteurs fonctionnant en ms (plage de fonctionnement de 1 à 32 767 ms). Le réglage du bit 15 sur 1 permet d'interrompre la décrémentation.	S et U
%SW80	Etat de l'E/S de base	**Pour le module analogique standard, %SW8x est décrit comme suit :** Bit [0] : toutes les voies analogiques en fonctionnement normal Bit [1] : module en phase d'initialisation Bit [2] : défaut d'alimentation Bit [3] : défaut de configuration Bit [4] : conversion de la voie d'entrée 0 en cours Bit [5] : conversion de la voie d'entrée 1 en cours Bit [6] : paramètre de la voie d'entrée 0 non valide Bit [7] : paramètre de la voie d'entrée 1 non valide Bit [8 et 9] non utilisé Bit [10] : dépassement par valeur supérieure de la voie d'entrée 0 Bit [11] : dépassement par valeur supérieure de la voie d'entrée 1 Bit [12] : dépassement par valeur inférieure de la voie d'entrée 0 Bit [13] : dépassement par valeur inférieure de la voie d'entrée 1 Bit [14] non utilisé Bit [15] : paramètre de la voie de sortie non valide	S

Mots système	Fonction	Description	Contrôle
%SW80	Etat de l'E/S de base	**Pour le module analogique standard TWDAMI4LT, %SW8x est décrit comme suit :** Bit [0 et 1] : état de la voie 0 0 0: Voie analogique en fonctionnement normal 0 1: paramètre de la voie d'entrée non valide 1 0: valeur d'entrée indisponible (module en phase d'initialisation, conversion en cours) 1 1: valeur de la voie d'entrée non valide (dépassement par valeur inférieure ou supérieure) Bit [2 et 3] : état de la voie 1 (même description que le bit [0 et 1]) Bit [4 et 5] : état de la voie 2 (même description que le bit [0 et 1]) Bit [6 et 7] : état de la voie 3 (même description que le bit [0 et 1]) Bit [8 et 15] : non utilisés	S
%SW80	Etat de l'E/S de base	**Pour le module analogique standard TWDAMI8HT, %SW8x est décrit comme suit :** Bit [0 et 1] : état de la voie 0 0 0: Voie analogique en fonctionnement normal 0 1: paramètre de la voie d'entrée non valide 1 0: valeur d'entrée indisponible (module en phase d'initialisation, conversion en cours) 1 1: valeur de la voie d'entrée non valide (dépassement par valeur inférieure ou supérieure) Bit [2 et 3] : état de la voie 1 (même description que le bit [0 et 1]) Bit [4 et 5] : état de la voie 2 (même description que le bit [0 et 1]) Bit [6 et 7] : état de la voie 3 (même description que le bit [0 et 1]) Bit [8 et 9] : état de la voie 4 (même description que le bit [0 et 1]) Bit [10 et 11] : état de la voie 5 (même description que le bit [0 et 1]) Bit [12 et 13] : état de la voie 6 (même description que le bit [0 et 1]) Bit [14 et 15] : état de la voie 7 (même description que le bit [0 et 1])	S
%SW80	Etat du port CANJ1939 Twido Extreme	**Pour Twido Extreme uniquement, %SW80 est décrit comme suit :** Bit [0] Erreur d'initialisation - Repère perdu à l'occasion d'une réclamation Bit [1] Erreur d'initialisation - Impossible de réclamer un repère Bit [2] Etat passif d'erreur sur le port Bit [3] Etat du bus désactivé sur le port	S

Mots système	Fonction	Description	Contrôle
%SW81	● Etat du module d'expansion d'E/S 1 : définitions identiques à %SW80 ● Etat du module maître CANopen à l'adresse d'expansion 1 : ● Bit [0] état de configuration (1 = configuration OK ; 0 = erreur de configuration) Bit [1] état Operational (1 = échange PDO ON ; 0 = échange PDO OFF) Bit [2] état Init (1 = état d'initialisation ON ; 0 = état d'initialisation OFF) Bit [3] instruction CAN_CMD terminée (1 = terminée, 0 = en cours) Bit [4] erreur avec l'instruction CAN_CMD (1 = erreur; 0 = OK) Bit [5] erreur d'initialisation (1 = erreur ; 0 = OK) Bit [6] perte de message, erreur d'alimentation (1 = erreur ; 0 = OK) **Remarque :** Pour le maître de bus CANopen intégré de Twido Extreme, le mot système spécifique réservé est toujours %SW81 (les mots %SW82 à %SW87 ne sont pas utilisés).		S
%SW82 (1)	Etat Module d'expansion d'E/S 2 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 2 : définitions identiques à %SW81 (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.		S
%SW83 (1)	Etat Module d'expansion d'E/S 3 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 3 : définitions identiques à %SW81		S
%SW84 (1)	Etat Module d'expansion d'E/S 4 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 4 : définitions identiques à %SW81		S
%SW85 (1)	Etat Module d'expansion d'E/S 5 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 5 : définitions identiques à %SW81		S
%SW86 (1)	Etat Module d'expansion d'E/S 6 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 6 : définitions identiques à %SW81		S
%SW87 (1)	Etat Module d'expansion d'E/S 7 : définitions identiques à %SW80 Etat du module maître CANopen à l'adresse d'expansion 7 : définitions identiques à %SW81 (1) Ce mot système n'est pas disponible pour l'automate Twido Extreme TWDLEDCK1.		S
%SW94	Signature de l'application	En cas de modification de l'application, en termes de configuration ou de programmation de données, la signature (somme de tous les checksum) change en conséquence. Si %SW94 = 91F3 en hexadécimal, la signature de l'application est 91F3 en hexadécimal. **Remarque :** La version du microprogramme doit être supérieure ou égale à 2.5.	S

Mots système	Fonction	Description	Contrôle

%SW96	Commande et/ ou diagnostic de fonction d'enregistrement et de restauration pour le programme d'application et %MW.	• Bit [0] : indique que les mots mémoire %MWi doivent être enregistrés dans l'EEPROM :
		  • à l'état 1 si une sauvegarde est requise ;
		  • à l'état 0 si la sauvegarde en cours n'est pas terminée.
		• Bit [1] : ce bit est défini par le microprogramme pour indiquer que l'enregistrement est terminé :
		  • à l'état 1 si la sauvegarde est terminée ;
		  • à l'état 0 si une nouvelle requête de sauvegarde est demandée.
		• Bit [2] : erreur de sauvegarde (reportez-vous aux bits 8, 9, 10 et 14 pour plus d'informations) :
		  • à l'état 1 si une erreur est apparue ;
		  • à l'état 0 si une nouvelle requête de sauvegarde est demandée.
		• Bit [6] : à l'état 1 si l'automate contient une application valide dans la RAM.
		• Bit [8] : indique que le nombre de %MW spécifiés dans %SW97 est supérieur au nombre de %MW configurés dans l'application :
		  • à l'état 1 si une erreur est détectée.
		• Bit [9] : indique que le nombre de %MW spécifiés dans %SW97 est supérieur au nombre maximum de %MW pouvant être définis par toute application dans TwidoSuite.
		  • à l'état 1 si l'erreur est détectée.
		• Bit [10] : différence entre la RAM interne et l'EEPROM interne (1 = oui) :
		  • à l'état 1 s'il y a une différence.
		• Bit [14] : indique si une erreur d'écriture sur l'EEPROM s'est produite :
		  • à l'état 1 si une erreur est détectée.
			S et U

%SW97	Commande ou diagnostic de fonction d'enregistrement et de restauration	Lors de la sauvegarde de mots mémoire, cette valeur représente le nombre physique %MW à enregistrer dans l'EEPROM interne. Lors de la restauration de mots mémoire, cette valeur est mise à jour avec le nombre de mots mémoire restaurés dans la RAM. Pour l'opération de sauvegarde, lorsque cette valeur est définie sur 0, les mots mémoire ne sont pas stockés. L'utilisateur doit définir le programme de logique utilisateur. Dans le cas contraire, le programme sera réglé sur 0 dans l'application de l'automate, sauf dans le cas suivant : Lors d'un démarrage à froid, ce mot est réglé sur -1 si l'EEPROM Flash interne ne possède pas de fichier mot mémoire **%MW** enregistré. Lors d'un démarrage à froid au cours duquel l'EEPROM Flash interne contient une liste de mots mémoire **%MW**, la valeur du nombre de mots mémoire enregistrés dans le fichier doit être écrite dans le mot système **%SW97**.	S et U

<table>
<tr><th>Mots système</th><th>Fonction</th><th>Description</th><th>Contrôle</th></tr>
<tr><td>%SW101
%SW102
(1)</td><td>Valeur de l'adresse Modbus du port</td><td>Lorsque le bit %S101 est paramétré sur 1, vous pouvez modifier l'adresse Modbus du port 1 ou du port 2. L'adresse du port 1 est %SW101, celle du port 2 est %SW102.
Remarque :
● En mode en connecté, il est impossible de modifier l'adresse du port 2 à l'aide du bit système %S101 et du mot système %SW102.
● %S102 et le port 2 ne sont pas disponibles pour l'automate Twido Extreme TWDLEDCK1.</td><td>S</td></tr>
<tr><td>%SW103
%SW104
(1)</td><td>Configuration pour l'utilisation du protocole ASCII</td><td>Lorsque le bit %S103 (Comm 1) ou %S104 (Comm 2) est réglé sur 1, on utilise le protocole ASCII. Le mot système %SW105 (Comm 1) ou %SW106 (Comm 2) doit être paramétré en fonction des éléments ci-dessous :

● Débit en bauds :
● 0: 1 200 bauds ;
● 1: 2 400 bauds ;
● 2: 4 800 bauds ;
● 3: 9 600 bauds ;
● 4: 19 200 bauds ;
● 5: 38 400 bauds.
● RTS/CTS :
● 0: désactivé ;
● 1: activé.
● Parité :
● 00: aucune ;
● 10: impair ;
● 11: pair.
● Bit d'arrêt :
● 0: 1 bit d'arrêt ;
● 1: 2 bits d'arrêt.
● Bits de données :
● 0: 7 bits de données ;
● 1: 8 bits de données.
Remarque : %S104, %SW104 et Comm 2 ne sont pas disponibles pour l'automate TWDLEDCK1 Twido Extreme.</td><td>S</td></tr>
</table>

Mots système	Fonction	Description	Contrôle
%SW105 %SW106 (1)	Configuration pour l'utilisation du protocole ASCII	Lorsque le bit %S103 (Comm 1) ou %S104 (Comm 2) est réglé sur 1, on utilise le protocole ASCII. Le mot système %SW105 (Comm 1) ou %SW106 (Comm 2) doit être paramétré en fonction des éléments ci-dessous : 15 14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 0 Timeout trame en ms | Délai de réponse en multiples de 100 ms **Remarque :** %S104, %SW106 et Comm 2 ne sont pas disponibles pour l'automate TWDLEDCK1 Twido Extreme.	S
%SW111	Etat de la liaison distante	Indication : le bit 0 correspond à l'automate distant 1, le bit 1 à l'automate distant 2, etc. Bit [0] à [6] : ● à l'état 0 = automate distant 1-7 absent ● à l'état 1= automate distant 1-7 présent Bit [8] à [14] : ● à l'état 0 = E/S distante détectée sur l'automate distant 1-7 ● à l'état 1 = automate d'extension détecté sur l'automate distant 1-7	S
%SW112	Code d'erreur de configuration ou de fonctionnement de la liaison distante	00: opérations réussies 01: expiration du timeout (esclave) 02: erreur de checksum détectée (esclave) 03: incohérence de configuration (esclave) 04 - (pour le port 1 uniquement) port indisponible, P-Unit connecté ou mode P-Unit Défini sur 1 par le système et doit être remis à zéro par l'utilisateur.	S
%SW113	Configuration de la liaison distante	Indication : le bit 0 correspond à l'automate distant 1, le bit 1 à l'automate distant 2, etc. Bit [0] à [6] : ● à l'état 0 = automate distant 1-7 non configuré ● à l'état 1 = automate distant 1-7 configuré Bit [8] à [14] : ● à l'état 0 = E/S distante configurée en tant qu'automate distant 1-7 ● à l'état 1 = automate d'extension configuré en tant qu'automate distant 1-7	S
%SW114	Activation des blocs horodateurs	Active ou désactive le fonctionnement des blocs horodateurs, par l'intermédiaire du programme utilisateur ou de l'afficheur. Bit 0 : 1 = active le bloc horodateur n°0 ... Bit 15 : 1 = active le bloc horodateur n°15 Au départ, tous les blocs horodateurs sont activés. Si les blocs horodateurs sont configurés, la valeur par défaut est FFFF. Si aucun bloc horodateur n'est configuré, la valeur par défaut est 0.	S et U

Mots système	Fonction	Description	Contrôle
%SW118	Mot d'état de la base automate	Affiche les défaillances détectées sur la base automate. Bit 9 : 0 = défaillance ou comm. externe Défaillance Bit 12 : 0 = RTC* non installé Bit 13 : 0 = défaillance de configuration (extension d'E/S* configurée, mais absente ou défaillante). Tous les autres bits de ce mot sont à l'état 1 et sont réservés. Pour un automate ne présentant aucune défaillance, la valeur de ce mot est FFFFh. **Remarque :** *Pour l'automate TWDLEDCK1 Twido Extreme, il n'existe aucune E/S d'extension et un RTC est intégré.	S
%SW120 (1)	Etat du module d'expansion d'E/S(2)	Un bit par module. Repère 0 = Bit 0 1 = Erreur 0 = OK	S
%SW121 %SW122	Taille de trame Modbus ASCII	Lorsque le bit %S103 (Comm 1) ou %S104 (Comm 2) est réglé sur 1, on utilise le protocole ASCII. Vous pouvez modifier la taille de trame Modbus ASCII du port 1 ou du port 2. La taille de trame Modbus ASCII et celle du port 2 est %SW122. La valeur est utilisée uniquement au démarrage de l'instruction EXCH. Ainsi, si certains bits sont déjà reçus, vous ne pouvez pas interrompre la réception jusqu'au dernier bit.	U

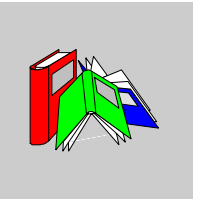
Note : (1) Ce mot système n'est pas disponible pour l'automate TWDLEDCK1 Twido Extreme.
(2) Si un seul module d'expansion manque à la mise sous tension, tous les bits de module d'expansion sont réglés sur 1 (Erreur).

Description des abréviations utilisées dans le tableau précédent

Tableau des abréviations :

Abréviation	Description
S	Contrôlé par le système
U	Contrôlé par l'utilisateur

Glossaire



!

% Préfixe qui identifie les repères de mémoire interne utilisés dans l'automate pour stocker les valeurs des variables, constantes, E/S, etc. du programme.

A

Adresse IP Adresse de protocole Internet. Adresse sur 32 bits affectée à des hôtes utilisant TCP/IP.

Adresse MAC Adresse de type Media Access Control (contrôle d'accès au support). Il s'agit de l'adresse matérielle d'un équipement. L'adresse MAC est affectée en usine à un module TCP/IP Ethernet.

Analyser le programme Commande permettant de compiler un programme et de rechercher les erreurs qu'il pourrait contenir : erreurs de syntaxe et de structure, symboles sans repère correspondant, ressources non disponibles que le programme tente d'utiliser et taille de programme trop importante pour la capacité de mémoire de l'automate. Les erreurs sont répertoriées dans le visualiseur des erreurs du programme.

Application Une application TwidoSuite est composée d'un programme, de données de configuration, de symboles et d'une documentation.

ASCII (American Standard Code for Information Interchange) Protocole de communication pour représenter les caractères alphanumériques, notamment les lettres, les chiffres et certains caractères graphiques et de contrôle.

Automate Automate programmable Twido. Il existe deux types d'automate : les automates compacts et les automates modulaires.

Automate compact Type d'automate Twido fournissant une configuration simple monobloc avec une expansion limitée. Les automates modulaires constituent l'autre type d'automate Twido.

Automate d'extension Automate Twido configuré en tant qu'esclave sur un réseau de liaison distante. Une application peut être exécutée dans la mémoire de l'automate d'extension et le programme peut accéder aux données d'E/S locales et d'expansion, mais les données d'E/S ne peuvent pas être transmises à l'automate maître. Le programme exécuté dans l'automate d'extension transmet des informations à l'automate maître à l'aide de mots réseau (%INW et %QNW).

Automate distant Automate Twido configuré pour communiquer avec un automate maître sur un réseau de liaison distante.

Automate maître Automate Twido configuré en tant que maître sur un réseau de liaison distante.

Automate modulaire Type d'automate Twido offrant une configuration flexible avec des possibilités d'expansion. Les automates compacts constituent l'autre type d'automate Twido.

Automate programmable Automate Twido. Il existe deux types d'automate : les automates compacts et les automates modulaires.

B

Bloc fonction Unité de programme comportant des entrées et des variables organisées pour calculer les valeurs des sorties à l'aide d'une fonction définie, telle qu'un temporisateur ou un compteur.

Blocs horodateurs Bloc fonction utilisé pour programmer les fonctions de réglage de la date et de l'heure afin de contrôler les événements. Nécessite l'option Horodateur (RTC).

Bobine Elément du schéma à contacts représentant une sortie de l'automate.

BootP Protocole basé sur UDP/IP (Protocole Bootstrap) permettant à un hôte d'amorçage de réaliser une configuration dynamique et sans aucune supervision de l'utilisateur. Le protocole BootP permet d'informer un hôte des adresses IP qui lui sont affectées.

Bus d'expansion Permet de connecter les modules d'expansion d'E/S à la base automate.

C

CAN — **CAN (Controller Area Network)** : bus de terrain développé à l'origine pour l'automobile qui est maintenant utilisé dans de nombreux domaines, de l'industrie au tertiaire.

Cartouche de mémoire — Cartouches de sauvegarde de mémoire en option permettant de sauvegarder et de restaurer une application (données de programme et de configuration). Deux tailles sont disponibles : 32 et 64 Ko.

Chargement automatique — Fonction constamment active permettant de transférer automatiquement une application depuis une cartouche de backup vers la RAM de l'automate en cas de perte ou d'altération de l'application. A la mise sous tension, l'automate compare l'application se trouvant dans sa RAM avec celle de la cartouche de backup de mémoire en option (si elle est installée). En cas de différence, l'application de la cartouche de backup est copiée dans l'automate et dans la mémoire EEPROM interne. Si aucune cartouche de backup n'est installée, l'application dans la mémoire EEPROM interne est copiée dans l'automate.

CiA — **CAN in Automation** : organisme international rassemblant les utilisateurs et constructeurs de produits CAN.

Client — Processus informatique nécessitant un service auprès d'autres processus informatiques.

COB — **COB (Communication OBject)** : unité de transport sur le bus CAN. Un COB est identifié par un identifiant unique codé sur 11 bits, [0, 2047]. Un COB contient au plus 8 octets de données. La priorité de transmission d'un COB est donnée par son identifiant, plus l'identifiant est faible plus le COB associé est prioritaire.

Commentaires — Textes que l'utilisateur saisit afin de donner des informations sur la finalité d'un programme. Pour les programmes en schéma à contacts, vous pouvez saisir jusqu'à trois lignes de texte dans l'en-tête réseau pour décrire la finalité du réseau. Chaque ligne peut contenir entre 1 et 64 caractères. Pour les programmes en liste d'instructions, vous pouvez saisir le texte sur une ligne de programme non numérotée. Les commentaires doivent être insérés entre parenthèses et astérisques, comme suit : (*INSEREZ LES COMMENTAIRES ICI*).

Commutateur — Equipement réseau connectant au moins deux segments de réseau distincts et permettant ainsi un trafic entre eux. Un commutateur détermine si une trame doit, selon son adresse cible, être bloquée ou transmise.

Compteur Bloc fonction utilisé pour compter les événements (comptage ou décomptage).

Compteurs rapides (FC) Bloc fonction proposant une fonction de comptage/décomptage plus rapide que celle du bloc fonction compteur. Un compteur rapide (FC) peut compter à une fréquence maximale de 5 kHz.

Compteurs très rapides (VFC) Bloc fonction proposant une fonction de comptage plus rapide que celle des blocs fonction compteur et compteur rapide (FC). Un compteur rapide (VFC) peut compter à une fréquence maximale de 20 kHz.

Concentrateur Equipement reliant plusieurs modules souples et centralisés afin de créer un réseau.

Constantes Valeurs configurées ne pouvant pas être modifiées par le programme en cours d'exécution.

Contact Elément du schéma à contacts représentant une entrée de l'automate.

D

Démarrage ou redémarrage à froid Démarrage de l'automate avec toutes les données initialisées sur les valeurs par défaut, le programme démarrant de zéro avec toutes les variables effacées. Tous les paramètres logiciels et matériels sont initialisés. Le chargement d'une nouvelle application dans la mémoire RAM de l'automate peut provoquer un redémarrage à froid. Un automate sans sauvegarde par pile démarre toujours à froid.

E

Editeur de configuration Fenêtre spécialisée de TwidoSuite permettant de gérer les configurations logicielles et matérielles.

Editeur de langage liste d'instructions Editeur de programmes simple permettant de créer et d'éditer un programme en liste d'instructions.

Editeur de langage schéma à contacts Fenêtre TwidoSuite spécialisée permettant d'éditer un programme en schéma à contacts.

Editeur de tables d'animation	Fenêtre spécialisée de l'application TwidoSuite permettant de visualiser et de créer des tables d'animation.
EDS	**Document de description électronique** : fichier de description de chaque équipement CAN (fourni par les constructeurs).
EEPROM	Mémoire morte effaçable et programmable électriquement (de l'anglais "Electrically Erasable Programmable Read-Only Memory"). Twido est doté d'une mémoire EEPROM interne et d'une cartouche de mémoire EEPROM externe en option.
Effacer	Commande qui permet de supprimer l'application de l'automate. Deux opérations sont possibles : ● Effacer le contenu de la RAM de l'automate, de l'EEPROM interne de l'automate et de la cartouche de backup en option installée. ● Effacer uniquement le contenu de la cartouche de backup en option installée.
En-tête réseau	Panneau apparaissant directement sur un réseau de schéma à contacts et pouvant être utilisé pour donner des informations sur la finalité de celui-ci.
Entrée à mémorisation d'état	Les impulsions entrantes sont capturées et enregistrées afin d'être analysées ultérieurement par l'application.
Etape	Une étape Grafcet désigne un état du fonctionnement séquentiel de l'automate.
Etat connecté	Etat de fonctionnement de TwidoSuite qui est affiché dans la barre d'état lorsqu'un PC est connecté à un automate.
Etat Init	Etat de fonctionnement de TwidoSuite affiché dans la barre d'état lorsque TwidoSuite est démarré ou qu'aucune application n'est ouverte.
Etat local	Etat de fonctionnement de TwidoSuite qui est affiché dans la barre d'état lorsque aucun PC n'est connecté à un automate.
Etats de fonctionnement	Indique l'état de TwidoSuite. Affiché dans la barre d'état. Il existe quatre états de fonctionnement : initial, local, connecté et sureveillance.
Executive Loader	Application Windows 32 bits permettant de décharger un nouveau microprogramme de l'automate vers un automate Twido.

F

Fichier d'application — Les applications Twido sont enregistrées dans des fichiers portant l'extension .twd.

FIFO — Premier entré, premier sorti (de l'anglais "First In, First Out"). Bloc fonction permettant de mettre les opérations en file d'attente.

Fonctions Date/ Heure — Fonctions permettant de contrôler les événements par mois, jour et heure. Voir "Blocs horodateurs".

Forçage — Attribution volontaire des valeurs 0 et 1 aux entrées et sorties de l'automate, même si les valeurs réelles sont différentes. Permet de déboguer un programme pendant son animation.

G

Gestionnaire de ressources — Composant de TwidoSuite qui surveille les besoins en mémoire d'une application lors de la programmation et de la configuration, en suivant les références aux objets logiciels faites par une application. Un objet est considéré comme étant référencé par l'application lorsqu'il est utilisé comme opérande dans une instruction de langage liste d'instructions ou dans un réseau de schéma à contacts. Affiche les informations d'état relatives au pourcentage de mémoire totale utilisée et émet un avertissement si l'espace mémoire disponible est insuffisant. Voir "Indicateur du bilan mémoire".

Grafcet — Permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.
Il s'agit d'une méthode analytique qui divise toute régulation d'automatisation en une série d'étapes auxquelles des actions, des transitions et des conditions sont associées.

H

Horodateur — Option permettant de maintenir une horloge à l'heure pendant une durée déterminée lorsque l'automate n'est pas sous tension.

Hôte — Nœud d'un réseau.

I

Indicateur du bilan mémoire Section de la barre d'état de la fenêtre principale de TwidoSuite qui affiche le pourcentage d'utilisation par une application de la mémoire totale de l'automate. Emet un avertissement lorsque l'espace mémoire disponible est insuffisant.

Initialiser (Init) Commande qui rétablit les états initiaux de toutes les valeurs des données. L'automate doit être en mode d'arrêt ou d'erreur.

Instance Dans un programme, objet unique qui appartient à un type précis de bloc fonction. Par exemple, dans le format de temporisateur %TMi, i est un nombre qui représente l'instance.

Instructions réversibles Méthode de programmation permettant de visualiser les instructions alternativement comme des instructions de liste d'instructions ou des réseaux de schéma à contacts.

Internet Interconnexion mondiale de réseaux de communication par ordinateur fonctionnant sur TCP/IP.

IP Protocole Internet (Internet Protocol). Protocole classique de la couche réseau. IP et TCP sont les plus utilisés.

L

Langage liste d'instructions Un programme écrit en langage liste d'instructions (IL) consiste en une série d'instructions exécutées de manière séquentielle par l'automate. Chaque instruction comprend un numéro de ligne, un code d'instruction et un opérande.

Langage schéma à contacts Un programme écrit en langage schéma à contacts consiste en la représentation graphique d'instructions d'un programme de l'automate, avec des symboles pour les contacts, bobines et blocs, sous la forme d'une série de réseaux exécutés de manière séquentielle par un automate.

Liaison distante Bus maître/esclave à haut débit conçu pour assurer l'échange d'une petite quantité de données entre un automate maître et un maximum de sept automates distants (esclaves). Deux types d'automate distant peuvent être configurés pour transférer des données vers un automate maître : un automate d'extension pour transférer les données d'application et un automate d'E/S distant pour transférer les données d'E/S. Un réseau de liaison distante peut comprendre des automates des deux types.

LIFO Dernier entré, premier sorti (de l'anglais "Last In, First Out"). Bloc fonction permettant d'effectuer des opérations de pile.

Lignes de commentaire Dans les programmes en liste d'instructions, les commentaires peuvent être saisis sur des lignes distinctes des instructions. Les lignes de commentaires ne portent pas de numéro de ligne et doivent être insérées entre parenthèses et astérisques, comme suit : (*INSEREZ LES COMMENTAIRES ICI*).

M

Masque de sous-réseau Masque de bit permettant d'identifier ou de déterminer les bits de l'adresse IP correspondant à l'adresse réseau et les bits correspondant aux portions du sous-réseau de l'adresse. Le masque de sous-réseau est constitué de l'adresse réseau et des bits réservés à l'identification du sous-réseau.

MBAP Protocole d'application Modbus (de l'anglais "Modbus Application Protocol").

Micropro-gramme de l'automate Système d'exploitation exécutant les applications et gérant les opérations de l'automate.

Modbus Protocole de communication maître-esclave permettant à un maître unique d'obtenir des réponses des esclaves.

Mode connecté Mode de marche de TwidoSuite dans lequel un PC est connecté à l'automate et dans lequel l'application contenue dans la mémoire du PC est identique à celle contenue dans la mémoire de l'automate. Le fonctionnement en ligne permet de déboguer une application.

Mode de scrutation Indique la façon dont l'automate scrute un programme. Il existe deux types de scrutation : le mode normal (cyclique), dans lequel la scrutation s'effectue en permanence, ou le mode périodique, dans lequel la scrutation ne s'effectue que pendant une durée limitée (dans une plage de 2 à 150 ms) avant de lancer la scrutation suivante.

Mode local Mode de marche de TwidoSuite dans lequel aucun PC n'est connecté à l'automate et dans lequel l'application contenue dans la mémoire du PC est différente de celle contenue dans la mémoire de l'automate. Le mode local permet de créer et de développer une application.

Mode Surveillance Etat de fonctionnement de TwidoSuite qui est affiché dans la barre d'état lorsqu'un PC est connecté à un automate dans un mode sans écriture.

Modules d'expansion d'E/S Les modules d'expansion d'E/S en option sont disponibles pour ajouter des points d'E/S à un automate Twido. (Certains modèles d'automate ne prennent pas en charge l'expansion.)

N

Navigateur d'application Fenêtre spécialisée de l'application TwidoSuite qui affiche l'arborescence graphique d'une application. Facilite l'affichage et la configuration d'une application.

Nœud Equipement adressable sur un réseau de communication.

O

Opérande Nombre, repère ou symbole représentant une valeur qu'un programme peut manipuler dans une instruction.

Opérateur Symbole ou code indiquant l'opération qu'une instruction doit réaliser.

P

Paquet Unité de données envoyée sur un réseau.

Passerelle Equipement reliant des réseaux dont l'architecture est différente et fonctionnant sur la couche application. Ce terme peut faire référence à un routeur.

Passerelle par défaut Adresse IP du réseau ou de l'hôte vers laquelle sont envoyés tous les paquets adressés à un réseau ou à un hôte inconnu. La passerelle par défaut est généralement un routeur ou un autre équipement.

PC Ordinateur personnel (de l'anglais "Personal Computer").

PLS Générateur d'impulsions. Bloc fonction qui génère une onde carrée avec des cycles d'activité de 50 % et d'inactivité de 50 %.

Point de réglage analogique Tension appliquée qui peut être réglée et convertie en une valeur numérique utilisable par une application.

Préférences Boîte de dialogue comprenant des options sélectionnables permettant de configurer les éditeurs de programmes en liste d'instructions et schéma à contacts.

Programmateur cyclique Bloc fonction dont le fonctionnement est semblable à celui des programmateurs cycliques électromécaniques : les modifications d'étapes sont associées aux événements externes.

Protection Se réfère aux deux types de protection d'une application : la protection par mot de passe, qui permet de contrôler l'accès à l'application, et la protection de l'application de l'automate, qui empêche la lecture et l'écriture sur un programme d'application.

Protocole Définit les formats de message et un jeu de règles utilisé par au moins deux équipements pour communiquer à l'aide de ces formats.

PWM Modulation de largeur (de l'anglais "Pulse Width Modulation"). Bloc fonction qui génère une onde rectangulaire avec un cycle d'activité variable pouvant être défini par un programme.

R

RAM Mémoire vive (de l'anglais "Random Access Memory"). Les applications Twido sont déchargées dans une mémoire RAM volatile interne afin d'être exécutées.

Redémarrage à chaud Après une coupure secteur, mise sous tension de l'automate sans modification de l'application. L'automate repasse à l'état dans lequel il était avant la coupure secteur et termine la scrutation qui était en cours. Toutes les données de l'application sont préservées. Cette fonction n'est disponible que sur les automates modulaires.

Références croisées Génération d'une liste d'opérandes, de symboles, de numéros de ligne/réseau et d'opérateurs utilisée dans une application pour simplifier la création et la gestion des applications.

Registres Registres spéciaux internes à l'automate dédiés aux blocs fonction LIFO/FIFO.

Repères Registres internes de l'automate permettant de stocker les valeurs des variables, constantes, E/S, etc. du programme. Le symbole de pourcentage (%) utilisé en préfixe permet d'identifier les repères. Par exemple, %I0.1 indique un repère dans la RAM de l'automate contenant la valeur de la voie d'entrée 1.

Réseau Equipements interconnectés partageant un chemin de données et un protocole de communication communs.

Réseau Un réseau est situé entre deux barres de potentiel d'une grille et se compose d'un groupe d'éléments graphiques reliés entre eux par des liaisons horizontales et verticales. Un réseau peut être constitué au maximum de sept lignes et onze colonnes.

Réseau schéma à contacts/liste d'instructions Affiche les parties d'un programme en liste d'instructions qui ne sont pas réversibles en langage schéma à contacts.

Routeur Equipement connectant au moins deux parties d'un réseau et permettant aux données de circuler entre ces deux parties. Un routeur examine chaque paquet reçu et décide s'il doit le bloquer pour l'isoler du reste du réseau ou s'il doit le transmettre. Le routeur tente d'envoyer le paquet sur le réseau en empruntant le chemin le plus efficace.

RTC De l'anglais "Real-Time Clock". Voir "Horodateur".

RTU De l'anglais "Remote Terminal Unit". Protocole utilisant huit bits permettant la communication entre un automate et un PC.

Run Commande permettant d'exécuter un programme d'application sur l'automate.

S

Sauvegarder Commande permettant de copier l'application contenue dans la RAM de l'automate à la fois dans la mémoire EEPROM interne de l'automate et dans la cartouche de backup de mémoire en option (si elle est installée).

Scrutation Un automate scrute un programme et effectue principalement trois fonctions de base. Il lit d'abord les entrées et place les valeurs correspondantes dans la mémoire. Il exécute ensuite le programme d'application, instruction par instruction, puis il stocke les résultats dans la mémoire. Il utilise enfin les résultats pour mettre à jour les sorties.

Serveur Processus informatique fournissant des services à des clients. Ce terme peut également désigner le processus informatique hébergeant le service.

Sortie réflexe En mode comptage, la valeur courante du compteur rapide (%VFC.V) est mesurée en fonction des seuils configurés afin de déterminer l'état des sorties dédiées.

Sorties seuil Bobines contrôlées directement par le compteur très rapide (%VFC) en fonction des paramètres choisis lors de la configuration.

Sous-réseau Réseau physique ou logique au sein d'un réseau IP, qui partage une adresse réseau avec d'autres parties du réseau.

Stop Commande permettant d'arrêter un programme d'application exécuté par l'automate.

Symbole Chaîne de 32 caractères alphanumériques maximum, dont le premier caractère est alphabétique. Les symboles permettent de personnaliser les objets de l'automate afin de faciliter la maintenance de l'application.

Symbole non résolu Symbole sans repère de variable.

T

Table d'animation Table créée dans un éditeur de langage ou dans un écran d'exploitation. Lorsqu'un PC est connecté à l'automate, la table d'animation permet de visualiser les variables de l'automate et de forcer leurs valeurs lors d'un débogage. Elle peut être enregistrée dans un fichier distinct portant l'extension .tat.

Table de symboles Table des symboles utilisés dans une application. La table est affichée dans l'éditeur de symboles.

TCP Protocole de contrôle de transmission (de l'anglais "Transmission Control Protocol").

TCP/IP Suite de protocoles composée du protocole de contrôle de transmission et du protocole Internet. Suite de protocoles de communication sur laquelle repose Internet.

Temporisateur Bloc fonction utilisé pour sélectionner une durée pour le contrôle d'un événement.

Trame Groupe de bits constituant un bloc TOR d'informations. Les trames contiennent des informations ou des données de contrôle de réseau. La taille et la composition d'une trame sont définies par la technique de réseau utilisée.

Twido Gamme d'automates Schneider Electric comprenant deux types d'automate (compacts et modulaires), des modules d'expansion permettant d'ajouter des points d'E/S et des options telles que l'horodateur, les communications, l'afficheur et les cartouches de sauvegarde de mémoire.

TwidoSuite Logiciel de développement graphique 32 bits fonctionnant sous Windows qui permet de configurer et de programmer des automates Twido.

Types de trame Ethernet II et IEEE 802.3 sont deux types de trame classiques.

U

UDP Protocole de communication (User Datagram Protocol) correspondant à la partie de la suite TCP/IP utilisée par des applications pour transférer des datagrammes. Le protocole UDP représente la partie TCP/IP responsable des adresses de port.

V

Validation auto par ligne Lors de l'insertion ou de la modification d'instructions en langage liste d'instructions, ce paramètre optionnel permet de valider les lignes de programme à mesure qu'elles sont saisies (recherche des erreurs et des symboles non résolus). Tous les éléments doivent être corrigés pour que le programmeur puisse quitter la ligne. Sélectionné à partir de la boîte de dialogue Préférences.

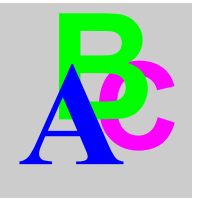
Variable Unité de mémoire pouvant être adressée et modifiée par un programme.

Variable de données Voir "Variable".

Visualiseur de références croisées Fenêtre spécialisée de l'application TwidoSuite permettant de visualiser les références croisées.

Visualiseur des erreurs du programme Fenêtre spécialisée de TwidoSuite permettant d'afficher les avertissements et erreurs du programme.

Index



Tri par symboles

-, 635
%Ci, 463
%DR, 520
%FC, 525
%INW, 43
%MSG, 541
%PLS, 517
%PWM, 514
%QNW, 43
%S, 674
%S0, 674
%S0=1
  Twido Extreme, 215
%S1, 674
  Twido Extreme, 215
%S10, 675
%S100, 680
%S101, 680
%S103, 681
%S104, 681
%S11, 675
%S110, 681
%S111, 681
%S112, 681
%S113, 681
%S118, 682
%S119, 682
%S12, 675
%S120, 682
%S121, 682
%S13, 675
%S17, 675
%S18, 675
%S19, 676
%S20, 676
%S21, 676
%S22, 676
%S23, 676
%S24, 677
%S25, 677
%S26, 677
%S31, 678
%S38, 678
%S39, 678
%S4, 674
%S5, 674
%S50, 678
%S51, 678
%S52, 679
%S59, 679
%S6, 674
%S66, 679
%S69, 679
%S7, 674
%S75, 679
%S8, 674
%S9, 675
  Twido Extreme, 215
%S95, 679
%S96, 680
%S97, 680
%SBR, 470
%SCi, 472

%SW, 683
%SW0, 683
%SW1, 683
%SW101, 694
%SW102, 694
%SW103, 694
%SW104, 694
%SW105, 695
%SW106, 695
%SW11, 684
%SW111, 695
%SW112, 695
%SW113, 695
%SW114, 695
%SW118, 696
%SW120, 696
%SW121, 696
%SW122, 696
%SW14, 685
%SW15, 685
%SW16, 685
%SW17, 685
%SW18, 685
%SW19, 685
%SW20..%SW27, 290, 685
%SW30, 685
%SW31, 686
%SW32, 686
%SW33, 686
  Twido Extreme, 332
%SW34, 686
%SW35, 686
%SW36, 686
%SW37, 686
%SW38, 686
%SW39, 686
%SW40, 686
%SW48, 686
%SW49, 687
%SW50, 687
%SW51, 687
%SW52, 687
%SW53, 687
%SW54, 687
%SW55, 687
%SW56, 687
%SW57, 687
%SW58, 687
%SW59, 688
%SW6, 683
%SW60, 688
%SW63, 688
%SW64, 688
%SW65, 689
%SW67, 689
%SW68, 690
%SW69, 690
%SW7, 684
%SW73, 690
%SW74, 690
%SW76, 690
%SW77, 690
%SW78, 690
%SW79, 690
%SW80, 690
  Twido Extreme, 332, 691
%SW81..%SW87, 289, 692
%SW94, 692
%SW96, 693
%SW97, 693
%TM, 460
%VFC, 528
*, 635
+, 635
/, 635

A

ABS, 635
Accès à la configuration
  PID, 586
Accès à la mise au point
  PID, 606
Accumulateur, 410
Accumulateur booléen, 410
ACOS, 639
Action dérivée, 628
Action intégrale, 627
Action proportionnelle, 626
Adressage de modules d'E/S analogiques, 155

Afficheur
correction de l'horodateur, 381
horloge calendaire, 380
ID et états de l'automate, 371
paramètres de port série, 379
variables et objets système, 373
vue d'ensemble, 368
Ajouter, 483
AND, instructions, 443
ASCII
communication, 74
communications, 110
configuration du port, 114
configuration logicielle, 113
configuration matérielle, 110
ASIN, 639
ATAN, 639

B

Backup et restauration
cartouche de mémoire étendue 64 Ko, 63
Bits mémoire, 27
Bits système, 674
BLK, 403
Bloc comparaison
élément graphique, 395
Bloc fonction compteur rapide (%VFC), 528
Bloc fonction compteur rapide (FC), 525
Bloc fonction d'échange, 541
Bloc fonction PLS
Twido Extreme, 195
Bloc fonction programmateur cyclique, 520
Bloc fonction PWM
Twido Extreme, 201
Blocs
dans des schémas à contacts, 390
Blocs comparaison, 391
Blocs fonction
compteurs, 463
dans une grille de programmation, 391
élément graphique, 395
Fonction pas à pas (%SCi), 472
programmateur cyclique, 520
programmation de blocs fonction standard, 453
PWM, 514
registre bits à décalage (%SBR), 470
registres, 509
temporisateurs, 455, 460
Blocs fonction avancés
objets mots et objets bits, 504
Blocs fonctions
blocs horodateurs, 547
présentation des blocs fonctions standard, 451
programmateur cyclique, 523
Blocs fonctions avancés
principes de programmation, 506
Blocs fonctions standard, 451
Blocs opération, 392
élément graphique, 395
Bobines, 390
éléments graphiques, 394
Boot-up, 257
Brochages
connecteur femelle du câble de communication, 79
connecteur mâle du câble de communication, 79
Bus AS-Interface V2
adressage automatique d'un esclave, 242
adressage des E/S, 245
changement d'adresse d'un esclave, 236
configuration logicielle, 227
description fonctionnelle générale, 221
diagnostic des esclaves, 235
échanges explicites, 246
échanges implicites, 245
écran de configuration, 225
en mode connecté, 233
esclave défectueux, 244
insertion esclave, 243
mise au point du bus, 238
mode de fonctionnement, 250
présentation, 220
principe de mise en œuvre logicielle, 224
prise en compte nouvelle configuration, 241
programmation et diagnostic du bus AS-Interface, 246
transfert de l'image d'un esclave, 239

Bus CANJ1939
méthode de configuration, 308
Bus CANopen
méthode de configuration, 270
Bus de terrain CANopen
échange explicite, 289
échange implicite, 288
programmation et diagnostic du bus de terrain CANopen, 289

C

Calcul, 483
CAN High (état haut), 254
CAN Low (état bas), 254
CAN, ligne de bus, 254
CAN_CMD, 292
CANJ1939
adresse source, 303
affectation d'adresse, 304
affichage des objets Transmettre/Recevoir, 322
base de connaissances, 299
boîtes de dialogue de configuration, 310
communication, 304
configuration d'élément, 310
configuration de la diffusion, 325
configuration de port, 310
configuration de réseau, 310
configuration du mode Expert, 329
création (ou suppression) des objets Transmettre/Recevoir, 314
création d'objets Transmettre/Recevoir, 315
détection de collision, 304
diffusion, 304
DP (page de données), 302
format PDU, 303
IDE (Extension d'identifiant), 303
messages de diffusion (recevoir), 326
messages de diffusion (transmettre), 325
mise en œuvre du bus CANJ1939, 305
mots système, 332
objets d'E/S, 330
PG (groupe de paramètres), 301
PGN (numéro de groupe de paramètres), 301
poste à poste, 304
PS (spécifique PDU), 303
requête SPN, 329
RTR (Requête de transmission distante), 303
SOF (début de trame), 303
SPN (numéro de paramètre suspect), 301
SRR (Substituer requête distante), 303
suppression d'objets Transmettre/Recevoir, 320
CANJ1939
identifiant, 302
CANopen
description, 254
Echange sous tension, 296
protocole, 254
Caractéristiques PID, 583
Chaînes de bits, 46
Champs de configuration d'entrée analogique
Twido Extreme, 180
Champs de configuration d'entrée TOR
Twido Extreme, 176
Champs de configuration d'une sortie TOR
Twido Extreme, 193
Commentaires de lignes Liste, 405
Communication avec un PC
à l'aide d'Ethernet pour Twido Extreme, 82
Communication par Ethernet, 76
Communication par modem, 76, 83
Communications
ASCII, 110
liaison distante, 96
Modbus, 123
Commutateur à clé
entrée, 170
Compteurs, 463
Programmation et configuration, 467
Configuration
PID, 586
port pour ASCII, 114
port pour Modbus, 127
table d'émission/réception pour ASCII, 115
Configuration d'entrée analogique
Twido Extreme, 178
Configuration d'entrée PWM
Twido Extreme, 182

Configuration d'un bloc fonction %PLS
Twido Extreme, 197
Configuration d'un bloc fonction %PWM
Twido Extreme, 204
Configuration de la diffusion
CANJ1939, 325
Configuration de la sortie hydraulique PWM
Twido Extreme, 209
Configuration de la sortie PWM
Twido Extreme, 200
Configuration des entrées TOR
configuration des entrées TOR, 172
Configuration des sorties TOR
Twido Extreme, 191
Configuration du mode Expert
CANJ1939, 329
Connexion Ethernet
Twido Extreme, 82
Conseils de programmation, 398
Contacts, 390
élément graphique, 393
Correction RTC, 546
COS, 639
Couche physique, 254
ligne de bus CAN, 254

D

Débordement, 485
index, 50
Débordement d'index, 50
Décrément, 483
DEG_TO_RAD, 641
Détection de fronts
descendants, 436
montants, 435
DINT_TO_REAL, 642
Diviser, 483
Documentation de votre programme, 405

E

E/S
repérage, 41
ECU (Bloc de commande électronique de l'anglais Electronic Control Unit), 299
Eléments CANJ1939, 299
Eléments de liaison
éléments graphiques, 393
Eléments graphiques
schémas à contacts, 393
Emission de message, 540
END_BLK, 403
En-tête réseau, 389
commentaires, 406
Entrée analogique
Twido Extreme, 178
Entrée analogique active
Twido Extreme, 178
Entrée analogique passive
Twido Extreme, 178
Entrée PWM
Twido Extreme, 182
Entrée TOR
Twido Extreme, 172
Entrées/sorties
Twido Extreme, 169
EQUAL_ARR, 659
Erreur, 485
EXCH, 540
Exemple
Compteur/Décompteur, 469
Exemple d'entrée analogique
Twido Extreme, 181
Exemple de configuration d'entrée PWM
Twido Extreme, 184
Exemple de configuration de la sortie PWM hydraulique
Twido Extreme, 216
EXP, 635
EXPT, 635

F

Facteur de correction de l'horodateur, 381
FIFO
fonctionnement, 511
introduction, 509
File d'attente, 509
Filtrage d'entrée
Twido Extreme, 172
FIND_, 661

Fonction pas à pas, 472
Fonctions horloges
horodatage, 549
réglage de la date et de l'heure, 551
vue d'ensemble, 546
Fonctions horodateurs
blocs horodateurs, 547
Forçage d'entrée
Twido Extreme, 172

G

Génération d'impulsions, 517
Grafcet
actions associées, 429
exemples, 422
instructions, 420
pré-traitement, 426
traitement séquentiel, 427
Grille de programmation, 388

H

Hydraulique
parasite, 211
rampe, 211

I

Incrément, 483
Index
objet de message, 319
Instruction EXCH, 540
Instructions
affectation, 441
AND, 443
arithmétiques, 483
chargement, 439
de comparaison, 481
de conversion, 490
END, 494
JMP, 497
logiques, 487
NOP, 496
NOT, 449
RET, 498
SR, 498
XOR, 447
Instructions arithmétiques, 483
Instructions booléennes, 435
explication du format utilisé dans ce manuel, 437
OR, 445
Instructions d'affectation, 441
numériques, 476
Instructions de comparaison, 481
Instructions de conversion, 490
Instructions de conversion entre mots simples et doubles, 492
Instructions de décalage, 488
Instructions de pile, 417
Instructions de saut, 497
Instructions de sous-programme, 498
Instructions en langage liste d'instructions, 411
Instructions END, 494
Instructions logiques, 487
Instructions numériques
affectation, 476
de décalage, 488
INT_TO_REAL, 642

J

JMP, 497

L

Langage liste d'instructions
vue d'ensemble, 408
Langages de programmation
présentation, 21
LD, 439
LDF, 436, 439
LDN, 439
LDR, 435, 439
Liaison ASCII
exemple, 120
Liaison distante
accès aux données E/S distantes, 101
communication, 74
communications, 96
configuration de l'automate distant, 99

configuration de l'automate maître, 99
configuration logicielle, 99
configuration matérielle, 97
exemple, 105
synchronisation de scrutation de l'automate distant, 100
Liaison Modbus
exemple 1, 135
exemple 2, 139
Life guarding, 263
Life-Time, 263
LIFO
fonctionnement, 510
introduction, 509
LKUP, 668
LN, 635
LOG, 635

M

MAX_ARR, 663
MEAN, 672
Mémoire
cartouche 64 Ko, 63
cartouche de 32 Ko, 60
sans cartouche, 58
structure, 54
Mémorisation d'entrée
Twido Extreme, 173
Messages de diffusion (recevoir)
CANJ1939, 326
Messages de diffusion (transmettre)
CANJ1939, 325
MIN_ARR, 663
Mise au point
PID, 606
Modbus
communication, 75
communications, 123
configuration du port, 127
configuration logicielle, 126
configuration matérielle, 123
esclave, 75
maître, 75
requêtes standard, 142
Mode
Operational, 259
pre-operational, 259
Modes de marche
Twido Extreme, 215
Modulation de la largeur d'impulsion, 514
Module analogique
exemple, 165
fonctionnement, 154
Module maître CANopen
adressage d'un PDO, 287
Modules analogiques
adressage, 155
configuration d'E/S, 156
Mots mémoire, 29
Mots système, 683
MPP, 417
MPS, 417
MRD, 417
Multiplier, 483

N

Network
Addressing, 43
Node guarding, 263
NOP, 496
NOP, instruction, 496
NOT, instruction, 449

O

Objet de message
index, 319
résumé, 318
Objets
blocs fonction, 44
flottant, 33
mot double, 33
mots, 29
objets bits, 27
structurés, 46
Objets bits, 504
adressage, 37
blocs fonction, 44
vue d'ensemble, 27

Objets flottants
adressage, 39
vue d'ensemble, 33
Objets mots, 504
adressage, 38
bloc fonction, 44
vue d'ensemble, 29
Objets mots doubles
adressage, 40
blocs fonction, 45
vue d'ensemble, 33
Objets Transmettre/Recevoir (CANJ1939)
affichage, 322
création, 315
création (ou suppression), 314
suppression, 320
OCCUR_ARR, 664
Onglet Animation
PID, 607
Onglet AT
PID, 598
Onglet Entrée
PID, 594
Onglet Général
PID, 587, 591
Onglet PID
PID, 596
Onglet Sortie
PID, 603
Onglet Trace
PID, 609
OPEN, 396
Opérandes, 410
OR exclusif, instructions, 447
OR, instruction, 445
OUT_BLK, 403
Ouverture de la table de configuration %PLS
Twido Extreme, 199
Ouverture de la table de configuration %PWM
Twido Extreme, 206

P

Paramètres, 456
Paramètres de contrôle
ASCII, 115
parasite, 211
Parenthèses
imbrication, 415
modificateurs, 415
utilisation dans des programmes, 414
PG (groupe de paramètres), 301
PGN
requête, 333
PGN (numéro de groupe de paramètres), 301
PID
configuration, 586
mise au point, 606
onglet Animation, 607
onglet AT, 598
onglet Entrée, 594
onglet Général, 587, 591
onglet PID, 596
onglet Sortie, 603
onglet Trace, 609
Vue d'ensemble, 579
Pile, 509
PLS
Twido Extreme, 194
Point de réglage, 150
Programmateurs cycliques
programmation et configuration, 523
Programmation
documentation de votre programme, 405
Programmation CANJ1939
messages d'erreur E/S, 332
recherche de données, 332
Programmation non réversible, 506
Programmation réversible, 506
Programme par schémas à contacts
conversion en liste d'instructions, 401
Programming Principles, 506
Protocole
Modbus TCP/IP, 75
Protocoles, 74

R

R, 441
Raccordement du câble de communication, 76
Racine carrée, 483
RAD_TO_DEG, 641
REAL_TO_DINT, 642
REAL_TO_INT, 642
Réception de messages, 540
Registre bits à décalage, 470
Registres
  FIFO, 511
  LIFO, 510
  programmation et configuration, 512
Réglage de boucle fermée, 623
Réglage de boucle ouverte, 624
Repérage
  objets indexés, 49
Repérage des E/S, 41
Repérage direct, 49
Repères d'E/S
  Twido Extreme, 168
Repères d'entrée PWM
  Twido Extreme, 182
Repères d'entrée TOR
  Twido Extreme, 174
Repères d'entrées analogiques
  Twido Extreme, 178
Repères de sortie PWM
  Twido Extreme, 200, 209
Repères de sortie TOR
  Twido Extreme, 191
Réseau schéma à contacts / liste d'instructions, 404
Réseaux
  inconditionnels, 404
Réseaux inconditionnels, 404
Réseaux langage schéma à contacts, 387
Reste, 483
Résumé
  objet de message, 318
RET, 498
Réversibilité
  introduction, 401
  recommandations, 403
ROL_ARR, 665
ROR_ARR, 665

S

S, 441
Sauvegarde et restauration
  cartouche de sauvegarde de 32 Ko, 60
  sans cartouche, 58
  structure de la mémoire, 54
Schémas à contacts
  blocs, 390
  éléments graphiques, 393
  introduction, 386
  OPEN et SHORT, 396
  principes de programmation, 388
SHORT, 396
SIN, 639
SORT_ARR, 667
Sortie du générateur d'impulsions
  Twido Extreme, 194
Sortie hydraulique PWM
  Twido Extreme, 209
Sortie PWM
  Twido Extreme, 200
Sortie TOR
  Twido Extreme, 191
Sorties PLS/PWM dédiées
  Twido Extreme, 194
Soustraire, 483
SPN (numéro de paramètre suspect), 301
SQRT, 635
SR, 498
ST, 441
STN, 441
SUM_ARR, 657
Symbolisation, 51

T

Table de contrôle
  Modbus, 129
Tables d'objets, 46
Tâches événementielles
  gestion des événements, 69
  les différentes sources d'événement, 67
  vue d'ensemble, 66
TAN, 639

TCP/IP
protocole, 75
Temporisateurs, 456
base temps de 1 ms, 461
introduction, 455
programmation et configuration, 460
TOF, type, 457
TON, type, 458
TP, type, 459
TOF, temporisateur, 457
TON, temporisateur, 458
TP, type de temporisateur, 459
Traitement numérique
Présentation, 475
TRUNC, 635
Twido Extreme
entrée analogique active, 178
entrée TOR, 172
entrées/sorties, 169
filtrage d'entrée, 172
forçage d'entrée, 172
mémorisation d'entrée, 173
repères d'E/S, 168
repères d'entrée TOR, 174
vue d'ensemble des entrées/sorties, 168
Twido Extreme
bloc fonction PLS, 195
bloc fonction PWM, 201
champs de configuration d'entrée analogique, 180
champs de configuration d'entrée TOR, 176
champs de configuration d'une sortie TOR, 193
configuration d'entrée analogique, 178
configuration d'entrée PWM, 182
configuration d'un bloc fonction %PLS, 197
configuration d'un bloc fonction %PWM, 204
configuration de la sortie hydraulique PWM, 209
configuration de la sortie PWM, 200
configuration des entrées TOR, 172
configuration des sorties TOR, 191
connexion Ethernet, 82
entrée analogique, 178
entrée analogique passive, 178
entrée PWM, 182
exemple d'entrée analogique, 181
exemple de configuration d'entrée PWM, 184
exemple de configuration de la sortie PWM hydraulique, 216
modes de marche, 215
ouverture de la table de configuration %PLS, 199
ouverture de la table de configuration %PWM, 206
parasite, 211
PLS, 194
rampe, 211
repères d'entrée PWM, 182
repères d'entrées analogiques, 178
repères de sortie PWM, 200, 209
repères de sortie TOR, 191
sortie du générateur d'impulsions, 194
sortie hydraulique PWM, 209
sortie PWM, 200
sortie TOR, 191
sorties PLS/PWM dédiées, 194
TwidoSuite
introduction, 20

V

Valeur absolue, 483
Validation d'objets, 26
Voie analogique, 152
Vue d'ensemble
PID, 579
Vue d'ensemble des communications, 74
Vue d'ensemble des entrées/sorties
Twido Extreme, 168

X

XOR, 447

Z

Zone d'action, 388
Zone de test, 388

You can download this technical publication and other technical information from our website at http://www.telemecanique.com.

Visit http://www.schneider-electric.com for your nearest Schneider Electric affiliate.

03/2007